AFRL-RI-RS-TR-2007-213
Final Technical Report
October 2007

COUNTERING INSIDER THREATS – HANDLING INSIDER THREATS USING DYNAMIC, RUN-TIME FORENSICS

PAR Government Systems Corp

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2007-213 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
ROBERT VAETH Work Unit Manager	WARREN H. DEBANY, Jr. Technical Advisor, Information Grid Division Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
OCT 2007

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
Apr 06 – Apr 07

4. TITLE AND SUBTITLE

COUNTERING INSIDER THREATS – HANDLING INSIDER THREATS USING DYNAMIC, RUN-TIME FORENSICS

5a. CONTRACT NUMBER
FA8750-06-C-0072

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)

Jason Hallahan

5d. PROJECT NUMBER
231G

5e. TASK NUMBER
JG

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
PAR Government Systems Corp
314 Jay St, Ste 1
Rome NY 13440-5600

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RIGB
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2007-213

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# AFRL/WS-07-2291

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The primary objective of this project was to research and develop applied computer forensic approaches for preventing and detecting insider threats in sensitive organizations in conjunction with advanced access control systems such as Fine-grained, Active, and Scalable Access Control (FASAC). Access Control is the fundamental basis of computer security, but still remains a relative weakness in dealing with everyday threats, especially those posed by insiders. To address the insider threats against critical information systems, an advanced access control approach was investigated that supports fine-grained, active, and scalable access control services.

15. SUBJECT TERMS
Access control, computer forensics, information systems, insider threat

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Robert Vaeth
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (*Include area code*)
U	U	U	UL	80	N/A

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Table of Contents

List of Figures

List of Tables

1. ABSTRACT

Access control is the fundamental basis of computer security, but still remains a relative weakness in dealing with everyday threats, especially those posed by insiders. Authentication, authorization, and audit are the three primary components of access control, which can be observed in countless mainstream implementations, including firewalls, virtual private networks, and file permissions. Virtually every security-related process or product is some flavor of access control. In Discretionary Access Control (DAC), the owner of an object can assign access to other users. In Mandatory Access Control (MAC), access is granted to users based on security policy. Unfortunately, current access control mechanisms are too coarse-grained, complex, and non-scalable to oppose the insider threat. Modern-day operating systems enforce access control at the granularity level of a file, but that does little to stop an insider who already has access to that file based on their position within the organization.

The insider threat is minimally addressed by current information security practices, yet the insider poses the most serious threat to the organization for various reasons. First, insiders are given a high level of trust. Second, it is easy for an insider to establish unauthorized entry points and anomalous channels into information systems. Third, more advanced forms of security such as encryption do not deal directly with the concept of access control. Fourth, current access control methods are too coarse-grained to look inside the box and prevent an insider from abusing his privileges. Finally, methods of auditing and forensics are generally after the fact and do little to prevent an insider from doing damage.

To address the insider threats against critical information systems, an advanced access control approach is needed that supports fine-grained, active, and scalable access control services. This will prevent the insider threats in terms of over-privileges based on the least-privilege principle, but cannot prevent the privilege-abuse problem. Applied computer forensic approaches are necessary to thwart the privilege-abuse problems where an insider does not have to violate access controls to perform malicious acts, as well as privilege escalation issues, where an insider would use various approaches to gain additional privileges such as root access. When used in combination, strong access control and applied computer forensics will serve to mitigate the threats posed by malicious insiders. The primary objective of this project was to research and develop applied computer forensic approaches for preventing and detecting insider threats in sensitive organizations in conjunction with advanced access control systems such as FASAC (Fine-grained, Active, and Scalable Access Control).

2. PROBLEM STATEMENT

The biggest modern-day threat against sensitive computer systems, networks, and data is the insider threat. An “insider” is an individual who possesses a certain level of access, privilege and trust within an organization due to their position, role, or task within that organization. The Department of Defense (DoD) defines an “insider” as anyone who uses authorized credentials to access a DoD computer and/or network, regardless of whether or not those credentials were acquired through legal channels. Whilst an outsider must gain access and privilege to a system using social engineering or some other method in order to damage that system, an insider generally inherits those capabilities by default. At this point the only thing that separates an insider employee from an outsider threat is their actions and intentions. Each insider poses the threat of malicious activity. We assume an insider is honest and is operating in the best interests of the organization. However, what if an insider's intentions change from benign to malicious? How do we detect malicious behavior from within our own walls? Modern-day computer defenses range from firewalls to intrusion detection systems (IDS) to access control lists (ACL), but their primary focus of mitigating the outsider threat remains the same. An insider is given a natural migration path inside of perimeter enterprise security controls. Efforts to incorporate these same defenses against insiders have thus far been fruitless. A great need still exists for a real-time, lightweight detection and mitigation system for insider misuse.

3. OBJECTIVE

The primary objective of this work was to focus on mitigating the insider threat against critical information systems using dynamic, run-time forensic investigation of insider usage based on anomalous behavior, alteration of system state, and misuse detection. The primary source of information for our investigative purposes was the Microsoft Windows file system and registry, which tracks and records many aspects of user and system behavior and usage.

4. INTRODUCTION

Access control and its component parts, authentication, authorization, and audit form the foundation of information security. Authentication establishes the identity of an individual wishing to access information assets. Authorization involves establishing what permissions or rights that individual has. Auditing involves gathering system usage data and user activity to discover potential security violations. Access control is the process of limiting access to information resources (objects) to authorized users, programs, processes, or other systems (subjects). Virtually every security-related process or product is some flavor of access control, including firewalls, virtual private networks, and file permissions.

Information security approaches have traditionally focused primarily on threats from outsiders, even though malicious insiders can present more severe threats to the enterprise. Insiders are individuals who already have certain privileges within a system based on their credentials. Existing access control approaches can be used for countering insider threats but still remain a weak link in dealing with insiders who might be over-privileged or who might abuse their privileges in a system or escalate their privileges beyond an appropriate level. Current access control mechanisms are too coarse-grained, static, and non-scalable to oppose the insider threats. In order to combat growing insider threats, access control methods must go above and beyond their current capabilities. FASAC is a proposed method to focus on mitigating insider threats against critical information systems using an advanced access control approach that supports fine-grained, active, and scalable access control services. Although FASAC could prove effective in terms of over-privileges based on the least-privilege principle and separation of duty, it cannot prevent the privilege-abuse problem. Applied computer forensics approaches are necessary to thwart privilege abuse problems and mitigate the threats posed by malicious insiders.

Recent intranet audits are showing an increased awareness of the insider threat, yet it still remains one of the largest risks to organizations today. An insider is someone who possesses the credentials to access internal systems without the scrutiny of many perimeter security devices such as firewalls and IDS. Due to insiders' easy entry and the inherent trust given to them, these individuals are the single greatest threat to organizations, enterprises, and governments. The largest threat posed by insiders today is the theft of proprietary data. Insiders are often compelled to steal organizational information assets for the purposes of espionage, job security, financial gain, and blackmail. The second biggest threat appears to be the installation of unauthorized software, including many types of harmful malware that can steal passwords and other information. The third biggest threat is insider leveraging of internal access to bypass existing security controls, such as tunneling through open ports on a firewall to create an anomalous channel for the purpose of information theft or unlawful disclosure. The insider threat is greater than ever given the increasing number of users, protocols, applications, and information assets. In addition, there is a philosophical disconnect between the user and the organization. The user desires the most liberal access to as much data and services as possible, while the organization would prefer a much more conservative model where the user is granted only the access needed to complete his job satisfactorily, the principle of least privilege. Although some security countermeasures address these individual threats, only access control and proactive forensics address the problem of insider misuse as a whole.

The need for digital forensics, which combines the concepts of digital evidence investigation with the legal system, has intensified over the past ten years as more commerce and information assets move into the digital realm. Most current forms of digital forensics are post facto attempts to prosecute corporate sabotage, espionage, computer theft, or misuse of corporate resources. The theft of proprietary information is the most common form of computer fraud. Today, digital forensics mostly involves confiscating a suspect's computer, preserving and duplicating its state, and then examining slack space on the hard disk, e-mails, deleted items, and changes to the registry and system files to try and establish culpability in a security incident. Unfortunately, current methods are more a form of recovery than prevention. Where an incident has already occurred, the damage has been done, but at least the perpetrator can be prosecuted. This project seeks to use real-time forensics to not only prove insider misuse but also prevent damage to the organization as a result.

The insider threat is minimally addressed by current information security practices yet the insider can damage an organization in so many ways and pose the most serious threat to the enterprise for various reasons. First, insiders are given a level of trust. Second, it is easy for an insider to establish unauthorized entry points and anomalous channels into information systems. Third, more advanced forms of security such as encryption do not deal directly with the concept of access control. Fourth, current access control methods are too coarse-grained to look inside the box and prevent an insider from abusing his privileges. Often the resulting damage from an incident in dollars and reputation is permanent, such as when an attacker exposes a bank database of credit card numbers. Traditional forensics and methods of auditing, which help companies identify and prosecute a criminal offender after the fact, are often of little consolation. Applied digital forensics, which monitor and audit computer systems in real-time, are a powerful preemptive strike against insider misuse. However, applying digital forensics in real-time is a daunting task, since there are so many files and processes to monitor, and the state of an average computer system or network is changing hundreds and even thousands of times per minute.

Before any real-time digital forensics can be applied to a system, there must be a clear determination of internal security controls and normal system behavior, as well as the files, processes, and behaviors that deserve the highest scrutiny. For instance, file deletion can be a benign act, but could also signal misuse, and should be monitored. System registries are often modified by software programs and system processes, but user modification of these files can signal suspicious behavior, such as the concealment of malicious activity. State changes of files with the attributes hidden or read-only, as well as the creation of these files, can also be considered suspicious depending on the context. The creation or modification of alternate data streams and metadata, as well as the usage of steganography, can also signal misuse.

User behavior and system usage patterns are central to applied digital forensics techniques. Over time, normal usage patterns can be established for a user (insider) just as they can for a network. Deviations from the normal patterns, called anomalies, can often signal misuse. User operations are often very consistent for a given context, and suspicious patterns are easily identifiable. Often, the operation is not as important as the order or context in which it appears. Order of operation is central to detecting misuse in real-time. For instance, deleting a file is not suspicious in and of itself, but a user who copies the contents of one file into another and then deletes the original file should draw suspicion. For another example, creating new directories is necessary of all system users, but navigation into a hidden system folder followed by the creation of a new directory should raise a red flag.

Designing an applied digital forensics approach that can identify all suspicious computer behavior in an organization is infeasible, just as it is impossible for the Transportation Security Administration (TSA) to comprehensively screen every passenger or piece of luggage at an airport. However, by applying some of the concepts discussed above, we can identify and monitor the “usual suspects” in terms of user behavior, operations, and execution patterns that are likely to signal insider misuse, thus greatly reducing the threat and frequency of damage.

5. RELATED WORK

It is necessary to define a new access control system, one that is scalable, active, and fine-grained. This system must be able to handle the security policy requirements of a large organization containing many decentralized and diverse users, while being easily managed. This system must be capable of adapting to the addition and deletion of new users, roles, permissions and operations as well as changing usage patterns and user behavior. This system must not become a performance bottleneck but must look deep enough into user operations to prevent misuse and inappropriate access on even the finest of levels.

A fine granularity is a very important characteristic for access control to possess. Typically, a large-scale application deals with a wide variety of contents to be shared. Some contents of an application might have multiple subfields or metadata requiring different levels of access control, depending on the sensitivity or classification of the individual fields. The control granularity of most existing access control mechanisms is at the level of the file, or even the directory, as opposed to data and information within the file, such as a record within a database. This coarse-grained control mechanism offers insufficient services for providing fine-grained protection of fields or contents within a file.

Current access control methods are simple to implement and effective in their task as long as the operating environment remains static. This is infeasible for a large, decentralized organization. In addition, most current methods make decisions on a predefined set of rules. Once a set of privileges is assigned to a user, that user can typically utilize those privileges in another context for any purpose desired. These context changes, such as task, end-system, location, and threat-level, should be considered; otherwise it is trivial for an insider to abuse his privileges. This is one of the most serious insider threats facing sensitive organizations. FASAC seeks to make access control decisions based not only on predefined rule-sets but also based on the context of the user. By changing the context of permissions when the context of the user changes, such as activating and deactivating permissions based on the current task of the user, the insider threat from privilege abuse is greatly minimized.

Reasonable scalability is not a feature of many existing access control approaches outside of Role Based Access Control (RBAC), as these systems typically become more and more complex and unmanageable as the number of roles grows. Unfortunately, scalability is necessary for large applications and especially critical if multiple organizations are involved in the same collaborative enterprise where some users may require various levels of functionality. Even RBAC can be difficult to manage when there exists a diverse set of roles and many unique users. There must be an efficient mechanism to deal with the complexity and large-scale demands of access control. FASAC seeks to be an access control approach that can handle a large-scale system that supports many users from many different organizations who may require different privileges under different contexts.

The primary focus of computer forensics is to obtain and preserve digital evidence. Most current forensic methods and tools assume that the collection, preservation and analysis of digital evidence usually will transpire after an incident. The following section provides a list of tools that are used mostly for this exact purpose: static, post-facto incident response and investigation. However, a primary goal of this project is to extend the functionality of one or more of these tools to the dynamic, run-time environment of a computer system or network for the purpose of insider threat mitigation and insider attack prevention. There are a number of freeware forensic tools that give researchers a good idea of what is out there for them to work with.

5.1 Windows Platform

The following applications serve different forensic purposes and run on Microsoft Windows Operating Systems based on the Windows NT kernel, such as XP SP2.

5.1.1 Cache Reader For Internet Explorer

http://www.wbaudisch.de/CacheReader.htm
This tool opens and reads the Windows system file index.dat which is stored in the Temporary Internet Files (TIF) folder for Internet Explorer (IE) version 5 and above. It displays, in either chronological or alphabetical order, the URLs of the pages stored in IE cache and the last date the site was visited by a system user. As a result, an examiner using this tool can gain insight into the browsing history and behavior of a target.

Cache Reader does not use Windows Application Programming Interfaces (APIs) and therefore it can show all entries, even those that are suppressed or lost by Internet Explorer and Windows Explorer. It can operate also on non-system folders, even on remote machines. Different from Explorer, it does not list the cookies that are not really contained in the TIF folder.

Searching for any text string and sorting is supported also. The cache index file of Internet Explorer is not changed at all, because Cache Reader exclusively uses the database of Internet Explorer.

Advantages:

- Forensically sound and does not change the integrity of the cache index file
- The user can view the index.dat file even on remote machines.
- The Cache Reader digs deeper into surfing history than a surface examination of browsing history would.

Disadvantages:

- This tool is non-transparent and cannot run in the background without target knowledge.
- Platform specific; it only works with Internet Explorer

5.1.2 Disk Investigator

http://www.theabsolute.net/sware/dskinv.html
Disk Investigator helps an examiner discover all that is hidden on a Microsoft Windows computer hard disk. It can also help to recover lost or deleted data as well as display the true drive contents by bypassing the operating system and directly reading the raw drive sectors. This program can also be used to view and search raw directories, files, clusters, and system sectors. Finally, an auxiliary use for this program is testing the effectiveness of file and disk wiping programs.

Advantages:

- Provides a thorough investigation of files residing on a WIN32 hard disk drive
- Freely available for download

Disadvantages:

- Cannot access remote networked hard disks
- Windows specific
- No source code available

5.1.3 Forensic Analyst's Software Tool (FAST)

https://sourceforge.net/project/showfiles.php?group_id=146246
FAST is a collection of software tools to perform forensic analysis on a WIN32 box. Currently, there are several components to the package, including:

1. Eindeutig – A tool for parsing Outlook Express database (DBX) files, which stores all e-mail accounts, addresses, contacts, and other personal information and communications.
2. Galleta – A tool for parsing IE cookie files.
3. Pasco – A tool for parsing IE Index.dat temporary internet files.
4. Rifiuti – A tool for parsing MS Windows Recycle Bin records.

Advantages:

- Platform Independent
- It is an open source program, and is freely available for download
- Combines investigation of both browsing and e-mail history

Disadvantages:

- Cannot access remote machines
- Designed for post-facto investigation

5.1.4 Evidor

http://www.x-ways.net/evidor/index-m.html
Evidor allows investigating parties to search text on hard disks and retrieve the context of keyword occurrences on computer media, not only by examining all files but the entire allocated space, even Windows swap/paging and hibernation files. Evidor also searches currently unallocated space and so-called slack space. That means it will even find data from files that have been deleted, if those files physically still exist.

Evidor is also an excellent tool for proving the presence or absence of confidential data on computer media, either to detect a security leak or confirm a lack thereof. With Evidor you often finds remnants or even intact copies of classified data that should have been encrypted, securely erased, or should not have existed on a media in the first place.

Advantages:

- Designed for use in sensitive environments with multiple layers of classification.

Disadvantages:

- Cannot access remote networked hard disks.
- Commercial product.

5.1.5 Gargoyle Investigator

http://www.000.shoppingcartsplus.com/catalog/item/1104418/619441.htm
Gargoyle is a software tool providing inspectors with the ability to conduct a quick search on a given computer or machine for known contraband and hostile programs. Gargoyle assists the investigator by providing a summary of installed programs, identification of potentially hostile or suspicious programs based on the loaded dataset, the classification of those hostile programs and the ability to ascertain incriminating behaviors or methods. The computer sophistication, covert behaviors, and paranoia levels can all be derived when searching for applications with a common theme. These behaviors can assist in assessing suspect capability, activities, intent, or threat.

Gargoyle quickly and easily determines whether malicious software (malware) is present on a system under investigation by employing custom datasets containing thousands of malware software signatures. Separate datasets can be created for various classifications of malware such as encryption software, steganography software, vulnerability assessment tools, network sniffers, port scanners, hacker tools, password cracking tools, and Denial of Service tools.

Advantages:

- One of the few programs that attempts to address the insider threat in a forensically sound manner.
- Attempts to classify user based on actions

Disadvantages:

- This tool is static; it does not detect the installation, execution or deletion of malware at run-time, but rather detects the presence of malware or remnants of past malware when a scan is performed.

5.1.6 StegAlyzerSS

http://www.sarc-wv.com/stegalyzerss.aspx

The Steganography Analyzer Signature Scanner (StegAlyzerSS) is a digital forensic analysis tool designed to extend the scope of traditional digital forensic examinations by allowing the examiner to scan files on suspect media for unique hexadecimal byte patterns (i.e., known signatures) left in files when particular steganography applications were used to embed hidden information within them.

StegAlyzerSS extends the signature scanning capability by also allowing the examiner to use the more traditional blind-detection technique for determining whether information may be hidden within potential carrier files. The program can scan the entire file system or individual directories and also has the capability to scan files from bit-by-bit (DD) images of suspect media for files that may contain known signatures of particular steganography applications.

Advantages:

- Has automated logging of key events and information of potential evidentiary value, a function that could be of great use if ported to real-time operation
- Forensically sound and can operate on image files, as well as a live system

Disadvantages:

- Signature-based. Will not detect recent attacks or new methods of information hiding.
- No remote scanning capabilities

5.1.7 Hurricane Search

http://www.hurricanesoft.com/prod01.htm
Formerly known as WinGREP, this is a fast, flexible search tool used to find data stored on computer hard drives and CD's. Hurricane Search has the power to search technical documents and product manuals from one environment. It can search Microsoft Word documents (.doc), Adobe PDF files, .zip and .jar files, as well as any binary files such as programming code.

Hurricane Search is a utility intended to make searching for words and strings quick and painless. Search results can be viewed in over IDEs, in the Hurricane Editor, or any other chosen editor. Hierarchical lists and Quick-Preview makes Hurricane Search fast and easy to use.

Advantages:

- Searches for words and strings in all types of binary files, even within compressed data and programming code
- Fast search and easy sorting

Disadvantages:

- Cannot be run without access to suspect computer

5.1.8 Decode

http://www.digital-detective.co.uk/freetools/decode.asp
Decode is a forensic date/time decoder designed to decode the various date/time values found embedded within binary and other file types. During a forensic examination, an investigator may need to decode a date or verify the date provided by other forensic software. This program can take a decimal value or a HEX value and convert it into a date & time in a variety of formats. Date and time values are stored within Windows in various formats. Internet History - index.dat, recycle bin INFO files, windows link files and Microsoft Office documents all contain a 64-bit date/time structure, while UNIX date/time structure is 32-bit. Unix format date & times appear quite often in binary files and plain text files. Some are stored in hexadecimal values or as a plain decimal value. The decimal format can be seen stored in many file types. Netscape 6+ history files store their date & times in the decimal format. This program can decode MAC times from all file types including those above, MS-DOS, and more.

Advantages:

- Freely available for download
- Can recognize many file types, which is a big positive for a forensic tool

Disadvantages:

- Source code not available

5.1.9 Active Ports

http://www.snapfiles.com/get/activeports.html
Active Ports enables an investigator to monitor all open TCP/IP and UDP ports on the local computer and also maps those ports to the owning application or process. It also displays a local and remote IP address for each socket connection and allows manual closing of any port. Active Ports can also help detect trojans and other malicious programs.

Advantages:

- Freeware
- Gives a network view of malicious activity and program execution.

Disadvantages:

- Is incompatible with host-based firewalls.
- Some anti-virus (AV) programs detect this program as a virus.

5.1.10 Filemon

http://www.microsoft.com/technet/sysinternals/FileAndDisk/Filemon.mspx

Filemon monitors and displays file system activity on local machines in real-time, as depicted in Figure 1, and is primarily used for showing how applications and processes use the files and Dynamic Link Libraries (DLLs), or tracking down problems in system or application file configurations. Filemon's includes a timestamping feature that displays when every open, read, write or delete, occurs as well as the outcome of the operation. The heart of Filemon is a file system driver that creates and attaches filter device objects to target file system device objects so that Filemon will see all IRPs and FastIO requests directed at local disk drives. When Filemon sees an open, create or close call, it updates an internal hash table that serves as the mapping between internal file handles and file path names. Whenever it sees calls that are handle based, it looks up the handle in the hash table to obtain the full name for display. If a handle-based access references a file opened before Filemon started, Filemon will fail to find the mapping in its hash table and will simply present the handle's value instead.

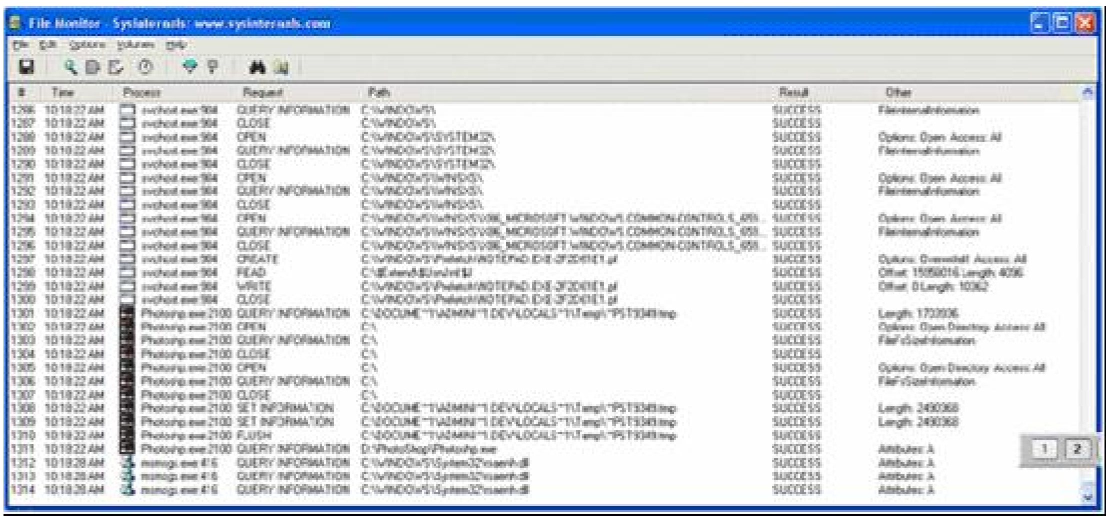


Figure 1 File Monitor

Advantages:

- Freely available
- Source code available for earlier versions
- Real-time monitoring and timestamping of file system

Disadvantages:

- Not designed for real-time forensics – no remote capabilities or remediation options
- No CLI mode
- Only available for Microsoft Windows Operating Systems.
- Limited filtering capabilities

5.1.11 Regmon

http://www.microsoft.com/technet/sysinternals/utilities/regmon.mspx

Regmon is a registry monitoring utility that shows which applications and processes are accessing the registry, which keys they are accessing, and the registry data that they are reading and writing - all in real-time. This utility goes beyond static registry tools to let investigators understand registry usage including how registry keys and values are being changed.

Regmon loads a device driver that uses a technique called system-call hooking. When a user-mode component makes a privileged system call, control is transferred to a software interrupt handler in NTOSKRNL.EXE, part of the Microsoft Windows OS kernel. This handler takes a system call number, which is passed in a machine register, and indexes into a system service table to find the address of the Windows function that will handle the request as shown in Figure 2. By replacing entries in this table with pointers to hooking functions, it is possible to intercept and replace, augment, or monitor Windows system services. Regmon hooks just the registry-related services but is merely one example of this capability in action.

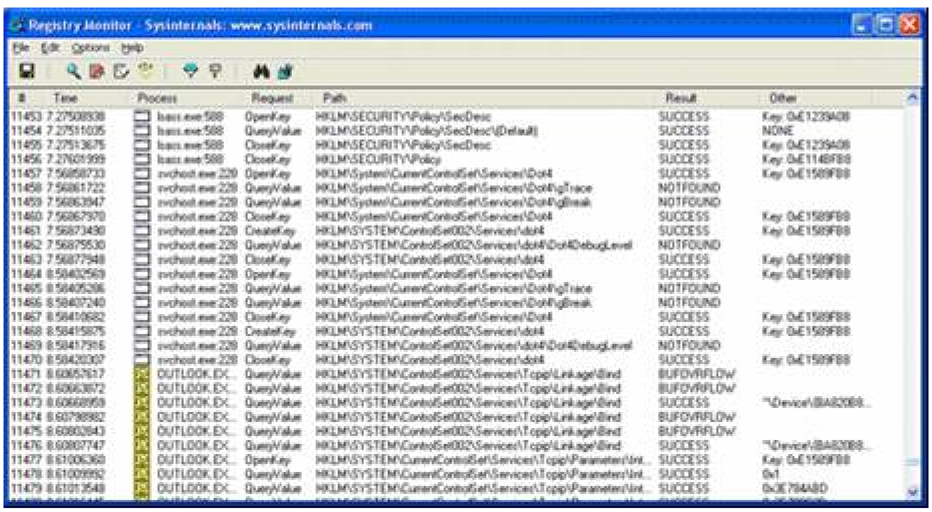


Figure 2 Registry Monitor

Advantages:

- Freely available
- Source code available for earlier versions
- Real-time monitoring and timestamping of the system registry

Disadvantages:

- Not designed for real-time forensics – no remote capabilities or remediation options
- No CLI mode
- Only available for Microsoft Windows Operating Systems
- Limited filtering capabilities

5.1.12Process Monitor

http://www.microsoft.com/technet/sysinternals/utilities/processmonitor.mspx

Process Monitor, depicted in Figure 3, is an advanced monitoring tool for Windows that shows real-time file system, registry and process/thread activity. It combines the features of *Filemon* and *Regmon*, and adds an extensive list of enhancements including rich and non-destructive filtering, comprehensive event properties such session IDs and user names, reliable process information, full thread stacks with integrated symbol support for each operation and simultaneous logging to a file, as depicted in Figure 4.

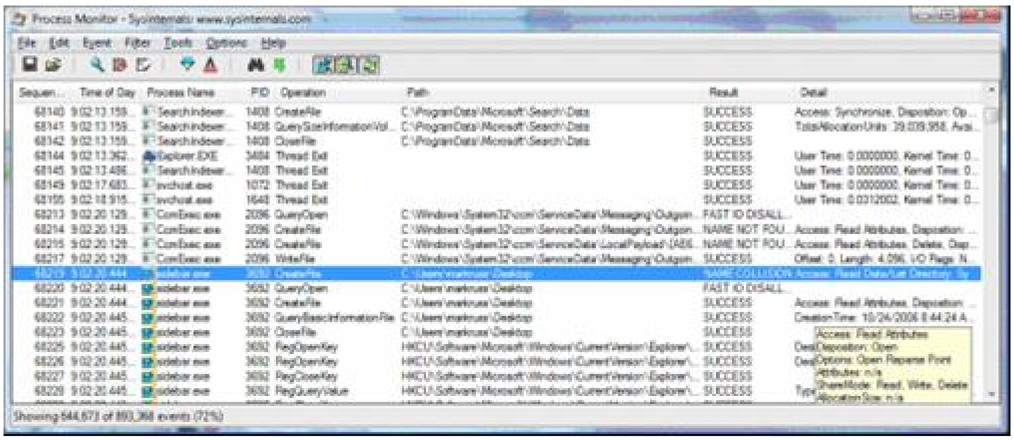


Figure 3 Process Monitor

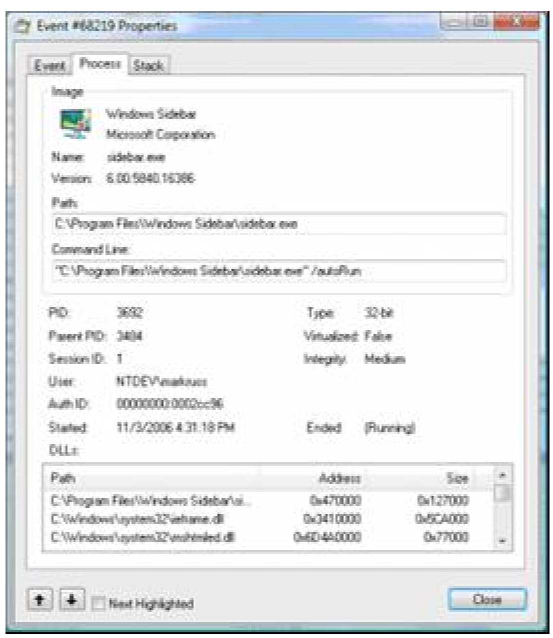


Figure 4 Event Properties

Advantages:

- Freely available
- Real-time monitoring and timestamping of file system, registry and process tables

Disadvantages:

- Not designed for real-time forensics – no remote capabilities or remediation options
- No CLI mode
- Only available for Microsoft Windows Operating Systems
- Limited filtering capabilities
- No source code available

5.2 UNIX Platform

The following applications serve various forensic needs and run on most operating systems based on the UNIX kernel.

5.2.1 The Sleuth Kit

http://www.sleuthkit.org/sleuthkit/desc.php
The Sleuth Kit is a collection of UNIX-based command line file and volume system forensic analysis tools that allows file system examination of a suspect computer in a non-intrusive fashion. Because the tools do not rely on the operating system to process the file systems, deleted and hidden content is shown. The Sleuth Kit tools can be run on a live UNIX system during Incident Response. These tools will show files that have been "hidden" by rootkits and will not modify the MAC times of files that are viewed. The Sleuth Kit can also display the details and contents of all Windows NT File System (NTFS) attributes including all Alternate Data Streams (ADS), display file system and meta-data structure details and finally create time lines of file activity, which can be imported into a spread sheet to create graphs and reports.

Advantages:

- Open source with source code freely available for modification.
- Written in C and PERL and runs on Linux, Mac OS X, OpenBSD, FreeBSD, Solaris, and CYGWIN.

Disadvantages:

- Limited automation so tedious human analysis is required
- Not as robust and comprehensive as comparable Windows tools

5.2.2 Chkrootkit

http://www.chkrootkit.org

Chkrootkit is a tool used to locally check for signs of a rootkit. It contains a shell script that checks system binaries for rootkit modification. It also checks if the network interface is in promiscuous mode as well as checks for lastlog deletions. The behavior detected by this program could either be as a result of a rootkit or perhaps a suspect trying to cover their tracks. For instance, this program also checks for sniffer logs, hidden processes that are running locally, and log deletion.

Advantages:

- Open source so the source code is freely available
- Works with most UNIX variants
- Can not only detect rootkits, but also suspicious behavior on a machine

Disadvantages:

- Static and must be run locally

5.2.3 Faust (File Audit Security Tookit)

http://security-labs.org/index.php3?page=faust
Faust is a Perl script that helps analyze files found after an intrusion or the compromising of a honeypot. Its goal is not to make the analysis, but to extract the pieces of information that the investigator will use afterward in their analysis. It is simply designed to gather the information, and display it in a way that helps the user to analyze a file. However, there are some patterns faust will seek such as email addresses, urls (http, ftp, https), IP addresses or references to critical UNIX directories such as /root. Faust can be effectively used to find rootkits, backdoors, and exploits running on a UNIX system.

Advantages:

- Open source; the source code is freely available
- Can detect rootkits and backdoors on a honeypot or real system
- Can be made network aware with remote capabilities using Perl or Python
- Very flexible and easily configured to what the examiner is looking for

Disadvantages:

- Static and must be run locally
- UNIX only

5.2.4 Coreography

http://www.engination.com/coreography/
Coreography is an open source utility for browsing memory images. It was originally intended as a tool for assisting in the analysis of core dumps. The tool has been expanded to parse any

Executable and Linkable Format (ELF) based memory image, including core dumps and ELF libraries, object files, executables, and even live processes.

With this utility, users are able to view segments of memory in their entirety or limited to selected parts. There is functionality to display all printable strings, or even to search for specific strings or any arbitrary data. Information learned from employing Coreography can be used, by itself, or in conjunction with similar utilities in the process of reverse engineering or a variety of other activities, such as malware detection.

Advantages:

- Can dynamically look at running processes, or run statically on memory dumps
- Open source; the source code is freely available
- Easily configurable

Disadvantages:

- Not designed for forensics
- Limited developer support

5.2.5 DCFL-DD

http://dcfldd.sourceforge.net/
Based on the DD program found in the GNU Coreutils package, DCFL-DD is often used to create bit-stream image files of media as part of a forensic acquisition process. DCFL-DD is an enhanced version of dd with MD5 hashing and multiple output capability. DCFL-DD can hash the input data as it is being transferred, helping to ensure data integrity, and also can output to multiple files or disks at the same time.

Advantages:

- Can ensure preservation of forensic data
- Open source; the source code is freely available
- Versions available for WIN32 (DD for Windows) and UNIX

Disadvantages:

- Not designed for real-time forensics – no remote capabilities
- Has a complex command line interface (CLI)

6. CHALLENGES

The applications above are just a few examples of the many tools available to a forensic investigator and incident response teams. Unfortunately, most if not all, do not perform any function that allows a security expert to prevent an intrusion or a malicious act. The major challenge of this project was to morph the functionality of these post-facto investigation tools into software that can address the insider threat in real-time and help to mitigate malicious activities as they occur. Thus there were three major challenges of this project. First, to develop a set of tools and techniques to detect malicious insider behavior and actions at run-time as opposed to after the damage is done. Second, once malicious behavior is detected by a user, these tools and methods must be able to stop an attack as opposed to just logging the incident. In other words, prevention of the insider attack is the second challenge. Finally, the third challenge was to preserve any and all incriminating data generated in the events leading up to and during the attempted attack, so that proper legal and corrective action can be taken after the insider is thwarted.

In summary, the challenges of this project were to take some of the ideas and tools presented in this survey and apply them to the insider threat for the purpose of:

- Run-time operation and detection of insider attack or possible malicious behavior.
- Prevention of insider attack.
- Preservation of forensic data implicating suspect.

6.1 Possible Extensions

Techniques and software can be developed to leverage hooks into real-time information provided by the file system, system logs and registry of modern operation systems. Existing hardware and software platforms provide all the necessary information to address malicious insiders, but a framework tool must be developed to properly collect, analyze and act upon that information. The tool will monitor network and system usage for potential malicious activity, flag suspicious events for investigation, and reporting flagged events to a remote central monitoring station. The threatening insider actions that can be detected include:

- Anomalous behavior and abnormal system activity
- Attempts to circumvent auditing and logging functions
- Copying, deleting, moving and printing sensitive files
- Network interface or system hardware manipulation
- Removable media or transferring data using unauthorized channel
- Attempts to “anonymize” network activities and web browsing
- Complex, sophisticated search queries against internal databases
- Downloading data to external, removable drives
- E-Mail, file and system log deletion
- Frequent and seemingly excessive use of encryption
- “Need-to-know” violations and privilege escalation attempts
- High volume printing
- Privilege escalation

Insider behavior can be observed using capabilities already included in modern operating systems and API hooks as well as sensors that have not yet been developed. This includes the file system, registry, system logs and ingress/egress network connections. Insider threat tools must monitor proactively in real-time and be effective within a large environment. The tools detect possible malicious activity, flag it, and then present the evidence and audit trail to human investigators. In addition, the insider threat solution must be developed to possess several additional characteristics, including the following:

Breadth – Large amount of information must be logged and monitored.

Covert – Insiders should know they are subject to monitoring but should not be aware of tools.

Extensible – Must be able to quickly address new threats and scenarios.

Fine-Granularity – Observing minute details just as important as "big picture."

Lightweight – Insider tools should use a minimal amount of local system resources.

Network Awareness – Investigators should be able to monitor insiders locally and remotely.

Platform-Independent – Should be reusable in different, heterogeneous environments.

Robust – Insider monitoring tools should not fail but if they do they should do so gracefully.

Scalable – Must be applicable in large, complex enterprises.

The bulk of this research will focus on utilizing programs such as Filemon, Regmon and Process Monitor, as well as system logs to monitor systems and uncover malicious insider behavior in real-time. Although these programs are intended for other uses, primarily system troubleshooting and process debugging, these programs provide a collection of capabilities that, if developed in combination with the characteristics listed above, could form a comprehensive forensic solution to the insider threat.

7. THREATS AGAINST COMPUTER SYSTEMS

Insiders pose a significant threat to any organization for a variety of reasons. First, insiders possess a trust level as a result of their role or task in the organization. Second, insiders have direct access in most cases to the systems that are most often attacked by outsiders. Finally, insiders have the ability to conduct operations on a computer or network that an outsider either does not have or must act diligently and intelligently in a malicious manner to obtain. Such abilities include:

- Physical access
- System logon
- Remote logon
- Firewall traversal
- Obtain IP address
- Browse WWW
- Send e-mail

The higher the level of trust, access, and ability that an insider has gained, the more dangerous they become to an organization should their actions become accidentally or intentionally malicious. Malicious actions carried out by insiders are virtually identical to the outsider threats faced by organizations.

- Access unauthorized resources
- Copy, delete, or move files without permission
- Crack or change passwords
- Exploit system or network vulnerabilities
- Execute malicious software
- Escalate privilege
- Circumvent auditing
- Install Rootkit or trojan horse
- Leak sensitive information

The concept of defense-in-depth usually focuses its attention outwardly and often ignores the threats beneath its nose or already within its borders. A different approach is needed to address insider threats. One of those methods, using the Microsoft Windows Registry and its related Windows functionality to monitor insider behavior, is explored here.

7.1 Microsoft Windows Registry

The Windows Registry is a hierarchical database that stores system parameters, security information, program configuration settings and user profiles. There are five root keys that cover different aspects of system operation. Each root key is comprised of many registry keys and their corresponding values. The Windows Operating System and applications frequently query the values of specific registry keys. The result of the query dictates system operation, as well as the user environment. This process occurs hundreds of times each second on a typical system. registry keys are also frequently added to the databases as new applications, users, and information are added to the system.

There are five components to every registry query discussed in Table 1 and depicted in Figure 5. These components include the name of the system process querying the registry, the type of query, the actual registry key being accessed, the status of the query, and the resultant value, if any.

Table 1 Components of a Registry Query

System Process	Query Type	Registry Key	Query Status	Result (Value)
Explorer.exe	OpenKey, QueryKey, CreateKey, QueryValue, SetValue, CloseKey	HKCU\Applications\Regmon.exe	SUCCESS, NOT FOUND, BUFFER OVERFLOW	0x200001A0

<u>Example</u>

Process: Explorer.exe:2876
Query: OpenKey
Key: HKCU\Software\Microsoft\Windows\ShellNoRoam\MUICache
Response: SUCCESS
Result: "Sysinternals Registry Monitor"

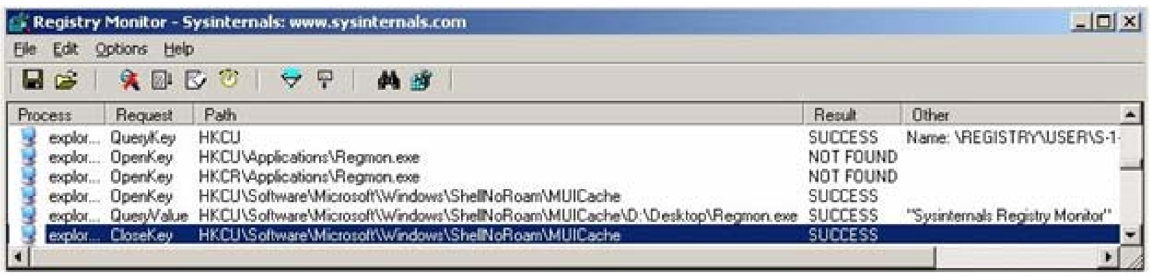


Figure 5 Sysinternals Registry Monitor

Under typical usage, a system user will perform a similar set of tasks and access certain programs consistently. Thus, registry activity is fairly predictable and a good source for detecting changes in user behavior or system use. Because of this fact it is fairly trivial to identify what registry keys are possibly malicious or can be used for a malicious purpose. Many of those keys are discussed throughout this report. The goal of this work was to use the power and predictability of the Windows Registry to perform real-time forensics and mitigate the insider threat. By focusing on registry accesses and changes to the registry, as well as comparing them to a normal usage patterns, malicious activity on a system, either purposeful or accidental, can be detected with reasonable assurance. The first research aim was to identify what registry keys are most often accessed and changed during malicious user or system behavior. The second aim was to identify how the registry is manipulated during specific attack scenarios, such as deleting sensitive files or spreading malicious code, as well as how the registry activity associated with these activities deviates from daily, average usage. The final aim was to develop a method for run-time prevention of the malicious actions by the user without destroying the digital evidence of those actions so that these individuals can be appropriately punished. The challenge remaining is the forensically-sound, real-time collection and analysis of hundreds of registry operations per second (on a typical system in use) for the purpose of insider misuse prevention.

7.1.1 Registry Operation

The Windows Registry is a central repository for literally all the configuration data (and more) on a Windows system. The registry was introduced in its current form in Windows 9x/ME and exists in all derivations and iterations of Microsoft Windows released since then. The Windows system (kernel), its users, applications and hardware all make use of the registry for their operation.

There are two parts to the Windows Registry, or at least two ways of looking at it. First, when a user executes regedt32.exe on a Windows system (regedit.exe on Win2K and previous) they see the logical representation of the registry in a hierarchical manner, as shown below in Figure 6.

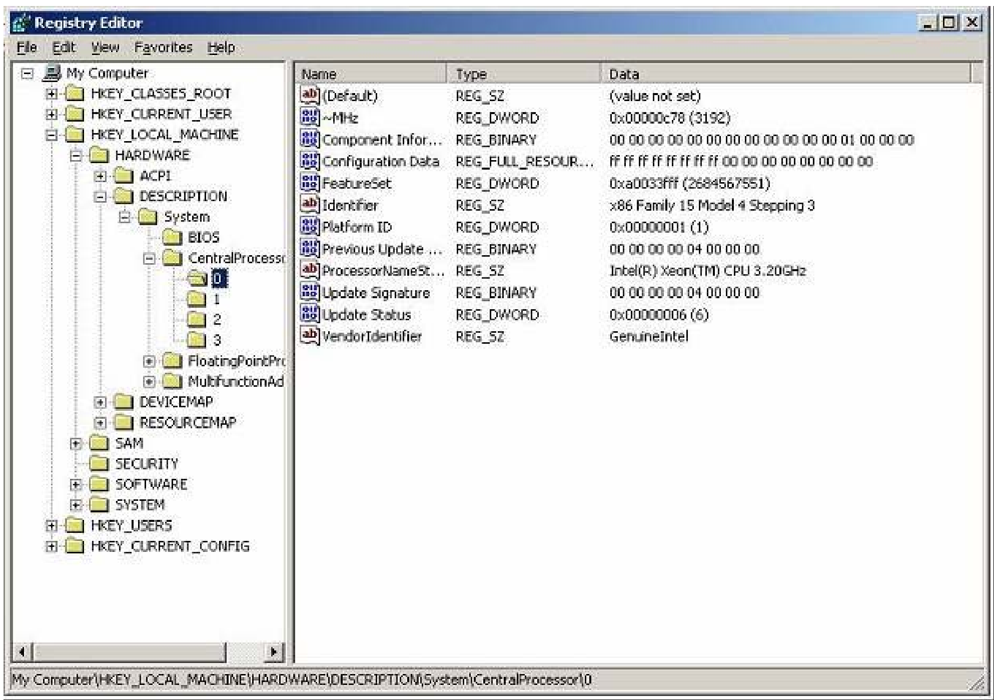


Figure 6 Registry Editor

The left pane shows all of the registry keys with the related subkeys beneath them while the right pane shows the values of the keys. There are five root keys in the Windows Registry as shown in Table 2:

Table 2 Windows Registry Root Keys

Root Key Name	Root Key Abbreviation
HKEY_CLASSES_ROOT	HKCR
HKEY_CURRENT_USER	HKCU
HKEY_LOCAL_MACHINE	HKLM
HKEY_USERS	HKU
HKEY_CURRENT_CONFIG	HKCC

Microsoft Windows uses a symbolic link to connect one key to another with a different path. This allows the same key and its values to appear at two different paths. HKLM and HKU are the only root keys that Windows physically stores in files. HKCU is a symbolic link to subkeys in HKU while HKCR and HKCC are symbolic links to subkeys in HKLM.

The second view of the Windows Registry is the physical view. The registry editor only shows the logical structure of the registry. However, the registry is not stored in a single file but rather a collection of binary system files, called registry hives. Only the HKLM and HKU root keys have corresponding hive files since the other three are links to the primary two, yet none of the five root keys are directly associated to any hive file. The relationship between the registry and the hive files is shown below in Table 3.

Table 3 Registry and Hive Files Relationships

Registry Path	Registry Hive	Physical Path
HKLM/SAM	SAM, SAM.LOG	C:/WINDOWS/SYSTEM32/CONFIG/SAM
HKLM/SECURITY	SECURITY, SECURITY.LOG	C:/WINDOWS/SYSTEM32/CONFIG/SECURITY
HKLM/SOFTWARE	SOFTWARE, SOFTWARE.LOG, SOFTWARE.SAV	C:/WINDOWS/SYSTEM32/CONFIG/SOFTWARE
HKLM/SYSTEM	SYSTEM, SYSTEM.LOG, SYSTEM.SAV	C:/WINDOWS/SYSTEM32/CONFIG/SYSTEM
HKLM/HARDWARE	Dynamic Hive kept entirely in volatile memory.	Memory
HKLM/DEFAULT	DEFAULT, DEFAULT.LOG, DEFAULT.SAV	C:/WINDOWS/SYSTEM32/CONFIG/DEFAULT
HKU/SID	NTUSER.DAT	%USERPROFILE%/NTUSER.DAT
HKU/SID_CLASSES	USRCLASS.DAT, USRCLASS.DAT.LOG	/LOCALS../APP../MIC../WIN../USRCLASS.DAT

7.1.2 Registry Data Types

There are three registry data types: STRING, DWORD, and BINARY. String values are the most common values utilized in the Windows Registry and consist of plain readable text (plaintext). There are three types of strings used in the registry: REG_SZ, REG_EXPAND_SZ, and REG_MULTI_SZ. REG_SZ is generally a “YES” or “NO;” REG_EXPAND_SZ usually stores a variable; and REG_MULTI_SZ is used to store arrays of multiple strings. BINARY values are most commonly used with hardware and configuration settings and consist of binary data displayed in hexadecimal format. DWORD values are most commonly used with system policy settings, device drivers, and services. DWORD values differ from BINARY values in that the binary data that can be entered is limited to 32 bits (4 bytes) in length and it can be entered in hexadecimal or decimal format.

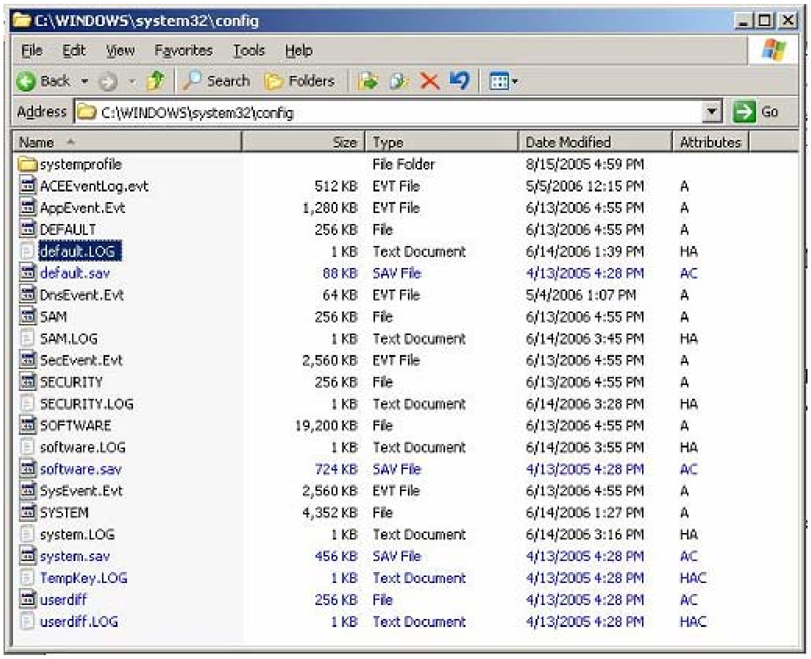


Figure 7 Registry Data Types

The actual hive file has no extension (e.g. DEFAULT), while a backup copy has a .sav extension (e.g. default.sav) and the transaction log of changes to a particular hive have a .log extension (e.g. default.LOG). The physical path for each hive file is shown above in Figure 7.

7.1.2.1 Forensic Application of Windows Registry

The Windows Registry stores and tracks many system and user operations that are pertinent to insider threat mitigation as well as real-time forensic investigations and include the following:

1. Hardware detected during Windows startup including external drives (e.g. USB Thumb Drives) and the resources associated with those devices. This information is stored in HKLM (HKEY_LOCAL_MACHINE) under the HARDWARE subkeys.

2. All information regarding Security Accounts Manager (SAM), which is the local security database containing information on all users and user groups as well as all associated security settings. This information is stored in HKLM under the SAM and SECURITY subkeys.

3. Settings for all applications installed on the computer, sorted by vendor, program, and version. This information is found in HKLM under the SOFTWARE subkeys.

4. Device driver settings and information on all system services, such as which ones are disabled or which ones run automatically upon system startup. This information is found in HKLM under the SYSTEM subkey.

5. Any program or component that is scheduled to run automatically during system startup (even hidden entries). These programs are found in HKLM/SOFTWARE/ Microsoft/Windows/CurrentVersion/Run.

6. HKCU/SOFTWARE/Microsoft/Windows/CurrentVersion/Explorer/ComDig32/LastVisted MRU is a registry key that maintains a list of files recently opened or saved via dialog boxes, excluding Microsoft Office files. File types tracked include .txt, .pdf, .htm, and .jpg as well as files opened or saved from within a web browser like Internet Explorer, Firefox and Opera. Related subkeys sort these files according to file extension.

7. HKCU/SOFTWARE/Microsoft/Windows/CurrentVersion/Explorer/RecentDocs is a registry key that maintains a list of files recently opened or executed using Windows Explorer (e.g. Windows Desktop, etc), including local or network files. Related subkeys sort these files according to file extension.

8. HKCU/SOFTWARE/Microsoft/Windows/CurrentVersion/Explorer/RunMRU is a registry key that maintains a list of entries executed using the “Run” command from the Windows Start Menu. Entries can be commands (e.g. regedt32, services.msc, cmd) or the full file path of an executed program.

9. HKCU/SOFTWARE/Microsoft/SearchAssistant/ACMru is a registry key that contains terms searched using Windows search, including folders, filenames, words and phrases.

10. HKCU/SOFTWARE/Microsoft/Windows/CurrentVersion/Uninstall is a registry key whose subkeys each represent a program installed on the computer, whether they are listed in the Add/Remove Programs component of the Control Panel or not. Information presented in the registry usually includes the install date, install source, and application version of the software in question.

11. HKLM/SYSTEM/CurrentControlSet/Enum/USBSTOR records the device serial number of any external USB storage devices, including memory cards, iPods, and thumb drives, that get mounted on the system.

12. HKLM/SYSTEM/CurrentControlSet/Services/ lists every Windows service that exists on the machine as well as its startup configuration and executable path. This is often where backdoors will hide.

13. HKLM/SYSTEM/CurrentControlSet/Services/Tcpip/Parameters/Interfaces/GUID is a registry key that contains recent network settings for each network adapter on the system, including IP address and default gateway.

14. HKLM/SOFTWARE/Microsoft/WZCSVC/Parameters/Interfaces/GUID is a registry key that, assuming the system is using Windows Zero Wireless Configuration Service, will show the last SSID that any wireless NIC was connected.

15. HKCU/SOFTWARE/Microsoft/Windows/CurrentVersion/Explorer/MapNetworkDriveMR U is a registry key that shows the last network mapped drive that computer was connected to, even if that connection has been discontinued.

16. HKCU/SOFTWARE/Microsoft/Windows/CurrentVersion/Explorer/UserAssist is a registry key whose subkeys maintain a list of system objects such as program, shortcut, and control panel applets that a user accessed.

17. HKCU/Software/Microsoft/ProtectedStorageSystemProvider is a registry key maintaining Windows Protected Storage. WPS is used to store passwords from Internet Explorer, Microsoft Outlook and Outlook Express as well as MSN Messenger.

18. HKCU/SOFTWARE/Microsoft/InternetExplorer/TypedURLs is a registry key listing the 25 most recent URLs that have been typed in the address bar of either Internet Explorer or Windows Explorer.

In summary, a great deal of information can be acquired from reading and analyzing the Windows registry. We have seen that this information includes:

- Applications, hardware and system services installed on the system and the settings associated with these entities in addition to which devices, programs and services run automatically at startup.
- Network settings, including the most recent connections made by wired and wireless adaptors.
- Internet Explorer browsing history.
- Recently accessed files, documents, and executables including the time of modification.
- Recent files, words and phrases searched by the user.
- User security settings and passwords
- Evidence of system alteration and concealment, including covert installation or execution of malicious code

Figure 8 shows the logical representation of some of the registry keys discussed above.

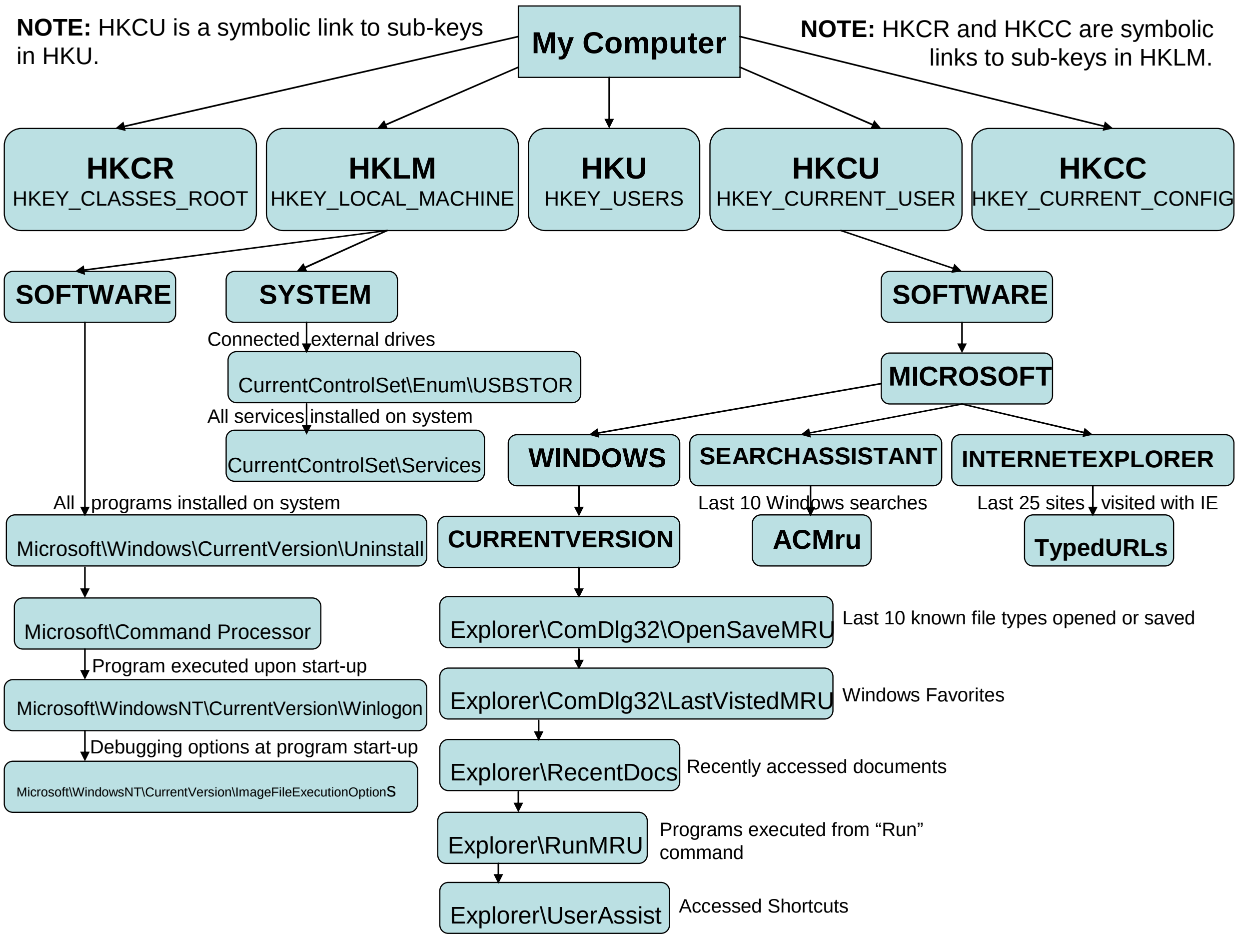


Figure 8 Registry Keys

7.1.2.2 Windows Registry Extensibility

The Windows Registry stores virtually all of the system’s configuration data, but can also be used to store a variety of data including passwords, text information, binary files and even executables. The registry also has a built-in feature that allows remote registry editing from other computers running Microsoft Windows. The structure or logical function of the registry cannot be manipulated.

New registry keys can be added and binary data can be entered as values for just about any registry sub key. In that same way that malware can be hidden inside of the registry by an attacker, the registry can be augmented by helpful code in the same manner.

Recently, security experts identified a vulnerability in the Windows operating system that could allow malware to hide within the Windows Registry using long string names. The weakness is caused by an error in the Windows Registry Editor Utility's handling of long string names and reportedly affects all versions of Windows 2000 and Windows XP. Therefore, a malicious program could hide itself in a registry key by creating a string with a long name, which would allow the malicious string and any created after it in the same registry key to remain hidden from view. This is particularly dangerous if this code exists in the "Run" registry keys. Malicious strings in this key will be executed whenever a user logs into an affected system.

7.1.2.3 Windows Registry Utilization Towards Real-Time Forensics

The Windows Registry will tell the tale of user and system activity on any Windows system. Preservation of registry data is easily achieved. By reconstructing a saved copy of a hive (e.g. DEFAULT.SAV) with the change transaction log (e.g., DEFAULT.LOG), the current state of the registry can be recreated or closely mirrored (e.g., DEFAULT with no extension). Furthermore, the transaction log will notify any monitoring program of changes to the registry. Therefore, malicious changes could be detected on the fly by any program that is developed to read and parse through the transaction logs related to each hive file.

The remote registry editing feature in Windows, depicted in Figure 9, has long been available but is rarely used and only vaguely understood, like the registry itself. Such a capability could be leveraged within a centralized environment government by Active Directory (e.g. Rome-2K domain at AFRL/RRS) in an effort to obtain a clearer picture of activity on a system. The remote registry service, shown in Figure 10, is intended to be used for remote assistance by IT staff so that they can access the Windows Registry on a networked system for the purpose of troubleshooting a hardware or software problem. If extended, this service would allow a centralized monitoring station to observe registry activity on all connected systems.

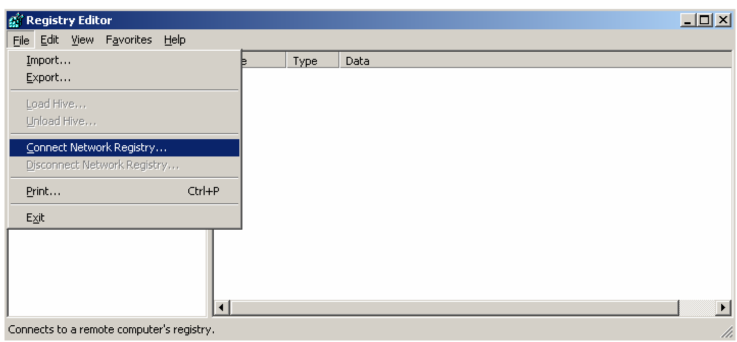


Figure 9 Registry Editor

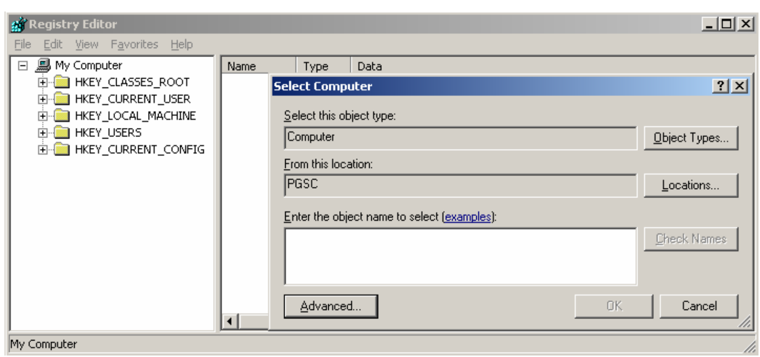


Figure 10 Registry Editor

7.1.2.4 Windows Registry Tools

The following tools provide different services related to the Microsoft Windows Registry. The registry can help identify the installation of a rootkit or other malicious software, as well as indicate recently executed files or altered system configuration.

MiTeC Windows Registry Analyzer – Windows Registry Analyzer is a tool for reading, viewing and analyzing the Windows Registry hive files in a forensic manner. It is free for both private and commercial use. No source code is readily available.

Offline Registry Parser – This tool, a Perl script, parses the registry file in binary mode, and prints out the keys with LastWrite times (in GMT format), as well as values, the data type of the value, and the data associated with the value.

AccessData Registry Viewer – This tool is a general registry viewer that can also access and decrypt protected storage data. It is a commercial product.

RootkitRevealer – This tool (Figure 11) is an advanced rootkit detection utility that runs on Microsoft Windows NT 4 and above.

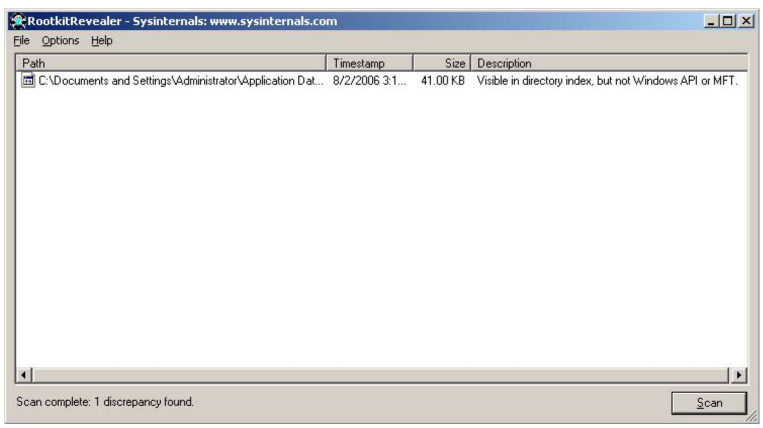


Figure 11 Rootkit Revealer

GMER **–** This is an application (Figure 12) that discovers hidden processes, hidden services, hidden files, hidden registry keys, and hidden drivers in an effort to unveil rootkits and other malicious software running on a system.

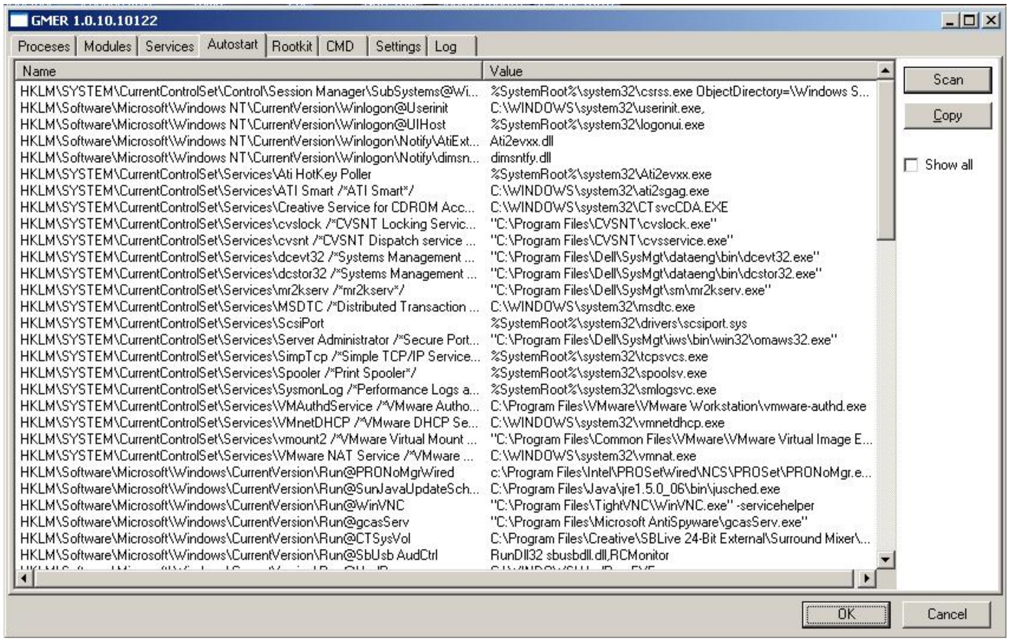


Figure 12 GMER

Hook Explorer – This is a small utility (Figure 13) designed to scan a target process and identify any user land hooks that may be installed by unknown code. It can tell the user if a file is hidden behind legitimate programs fooling existing firewall software.

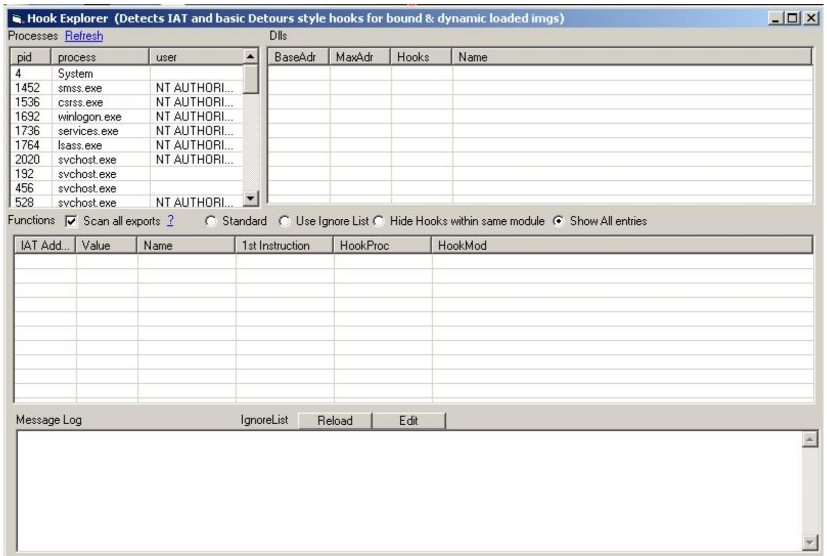


Figure 13 Hook Explorer

7.1.2.5 Windows Registry Dynamicism

Most of the information stored in the Microsoft Windows Registry is placed there automatically during the installation process of Microsoft Windows itself. Most other information in the registry comes from the installation programs of third-party applications and system hardware and changes made to the system by these programs and hardware. The registry can also be changed manually using the registry editor.

Specific registry keys are constantly accessed by the Windows OS as well as running programs, services, and processes. Virtually all WIN32 programs access, modify and use the registry to store information. Registry activity can vary from several accesses per minute to several thousand accesses per minute, depending on system use. Changes to the registry are less frequent and only occur when software or hardware makes changes to the system, whether automatically or through user interaction and input. Since the registry stores this information in files (hives), the hive files must be updated as frequently as the registry is changed.

For instance, when a system user is surfing the Internet using a web browser such as Internet Explorer, the registry keys related to Internet activity will constantly be updated to reflect the most recent websites visited, passwords and information entered, ActiveX controls and other material downloaded. Another example is if a user begins to load and unload software (e.g. Sniffer) on the system or mount and unmount hardware devices such as USB drives, the changes to relating registry keys will be frequent and significant.

As mentioned earlier, Sysinternals (http://www.sysinternals.com – now owned by Microsoft) has a freeware utility called Regmon which will display Windows Registry events, as depicted in Figure 14. Regmon is a registry monitoring utility that will show the user which applications are accessing the registry, which keys they are accessing, and the registry data that they are reading and writing - all in real-time. This advanced utility takes the user one step beyond what static registry tools can do, to allow users to see and understand exactly how programs use the registry. With static tools, a user might be able to see what registry values and keys changed. With Regmon that user will see how the values and keys changed. Regmon works on Windows NT/2000/XP/2003, Windows 95/98/Me and Windows 64-bit for Itanium and x64.

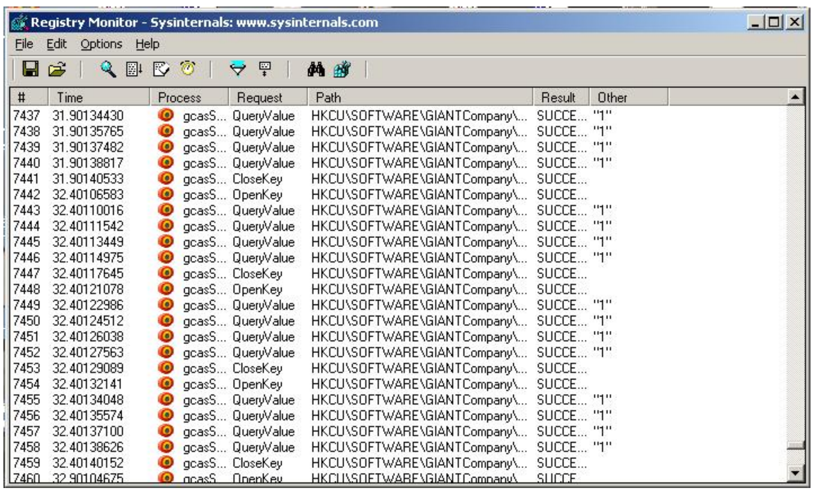


Figure 14 Registry Monitor

Spybot (http://www.safer-networking.org/) is a program designed to detect and remove spyware. It is freely available for download. Spybot has a helper application called TeaTimer, depicted in Figure 15, which flags registry changes as they happen and allows the user to permit or deny those changes. This makes it possible to monitor the registry and prevent unwanted changes, similar to the functionality of Microsoft's Anti-Spyware program.

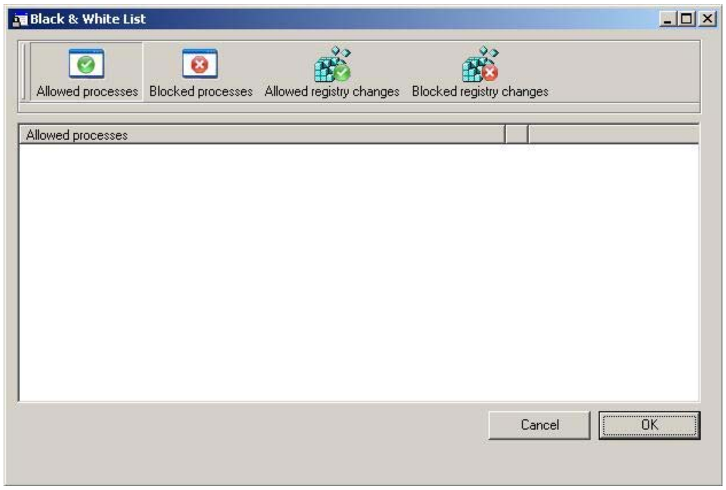


Figure 15 TeaTimer Black & White List

7.1.2.6 Windows Registry, Hives, Backups, and Log Files

The Registry Hives stored in files, (HARDWARE Hive is stored in RAM) which are DEFAULT, SAM, SECURITY, SOFTWARE, and SYSTEM, are updated to the latest version each time a Microsoft Windows system is shutdown. When the system is rebooted, the system uses the Hive file to initialize and load the system. Should startup fail, the system will revert back to a registry backup. Backups are performed periodically by the Windows system after some major system event (e.g. Installation of Microsoft Office) or manually by the user. While a Windows system is running, the Hive files are constantly being accessed, but they are not changed during system runtime. Any changes, modifications, additions or deletions to the registry are noted in the log files, which are updated very frequently (on the order of every few minutes). The next time the system is shutdown, changes in the log are written back into the Hive files, which are then used for the subsequent restart. Figure 16 below, illustrates the functionality of the Windows Registry.

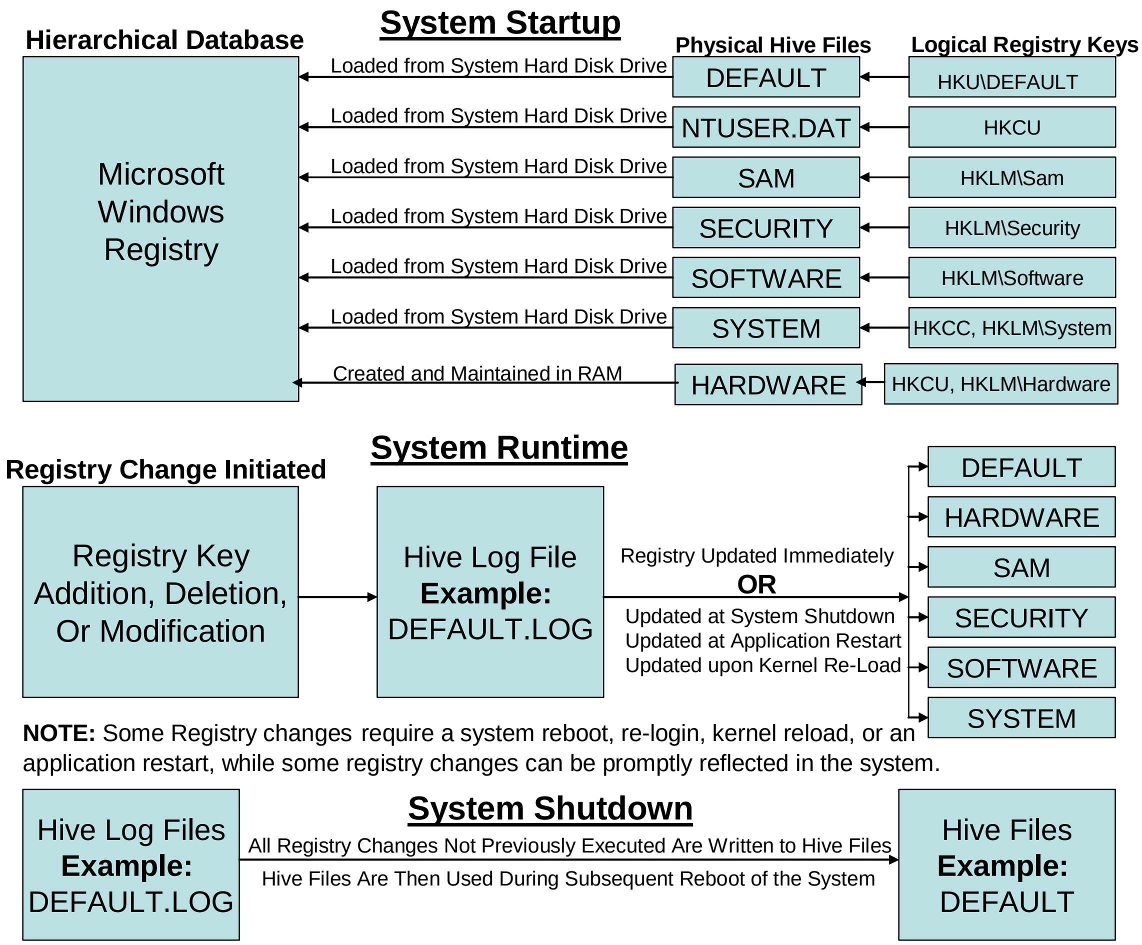


Figure 16 Windows Registry Functionality

8. MICROSOFT WINDOWS REGISTRY AND SYSTEM ATTRIBUTES

Edmond Locard's Exchange Principle, also known as Locard's Theory, states that "with contact between two items, there will be an exchange" and is applied to crime scenes in which the perpetrator(s) of a crime comes into physical contact with the scene. Locard postulates that the perpetrator(s) will both bring something into the scene and leave with something from the scene. Edmond Locard was the director of the very first crime laboratory in existence, located in Lyon, France. He later went on to say: "Physical evidence cannot be wrong, it cannot perjure itself, it cannot be wholly absent. Only human failure to find it, study and understand, it can diminish its value." While Locard's Principle does not apply directly to all cases of computer or electronic crime, the concept of the exchange of fragmentary evidence almost always applies.

Every action on a computer system spawns a reaction, usually in the form of several hundred state changes each minute and the creation or alteration of various attributes throughout the system. These attributes appear in physical, virtual and volatile memory, as well as system logs and registries. Actions taken on a MS Windows system usually precipitate hundreds of state changes within the registry. These changes are dynamic albeit somewhat predictable, which makes the Windows Registry a valuable repository for both real-time and post facto forensic evidence. Some scenarios and the related attributes are listed below.

8.1 Search Attributes

- Search
- Query
- Install

Example 1) Windows search for "Al Qaeda." A system user launches the integrated Windows Search functionality from the "Start" menu and searches for all files and folders on the system with the string "Al Qaeda."

Search term appears in **HKEY_CURRENT_USER\Software\Microsoft\Search Assistant\ACMru\5603** as well as **HKEY_USERS\S-1-5-21-214430146-809266577-2024027092-500\Software\Microsoft\Search Assistant\ACMru\5603** where **S-1-5-21-214430146-809266577-2024027092-500** is a unique system identity for a user. HKEY_USERS is a symbolic link to the entries in HKEY_CURRENT_USER root key for each unique system identity (system user).

The search strings in this example, "Al Qaeda," "default," and "lwip" appear as the data of the registry value in Figure 17 instead of the value itself simply due to the way Windows chooses to store the search history in chronological order ("000", "001", "002") so that older entries can be deleted in favor of new entries.

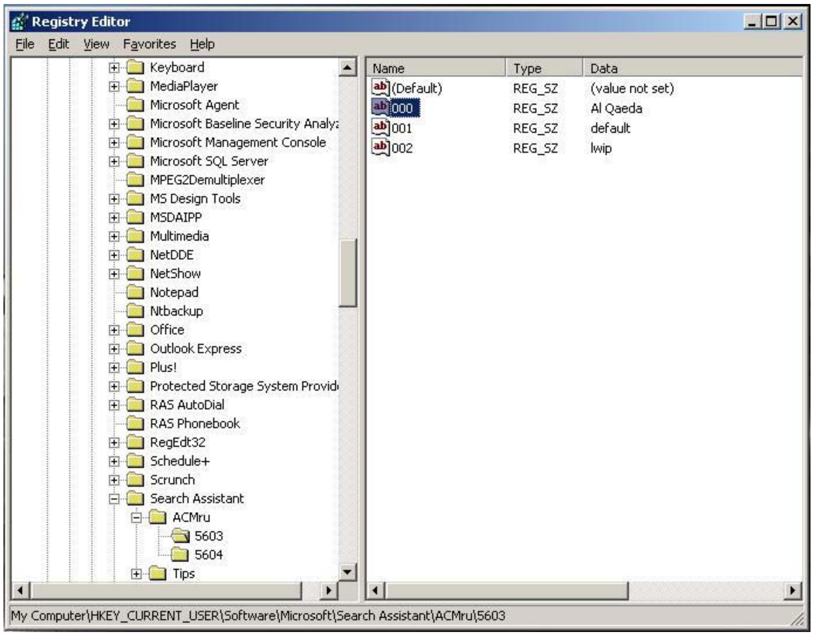


Figure 17 Google Search

Example 2) Google search for “Al Qaeda,” shown in Figure 18, within Internet Explorer using Google Toolbar. A system user launches Microsoft Internet Explorer and enters the string “Al Qaeda” within the Google Toolbar, which is a common search plug-in installed on most Windows system. Such an action will deliver the user to a web page with the most popular search results containing the string “Al Qaeda”.

Search term appears in **HKEY_CURRENT_USER\Software\Google\NavClient\1.1\History** as well as **HKEY_USERS\S-1-5-21-214430146-809266577-2024027092-500\Software\Google\NavClient\1.1\History** where **S-1-5-21-214430146-809266577-2024027092-500** is a unique system identity for a user. HKEY_USERS is a symbolic link to the entries in HKEY_CURRENT_USER root key for each unique system identity (system user).

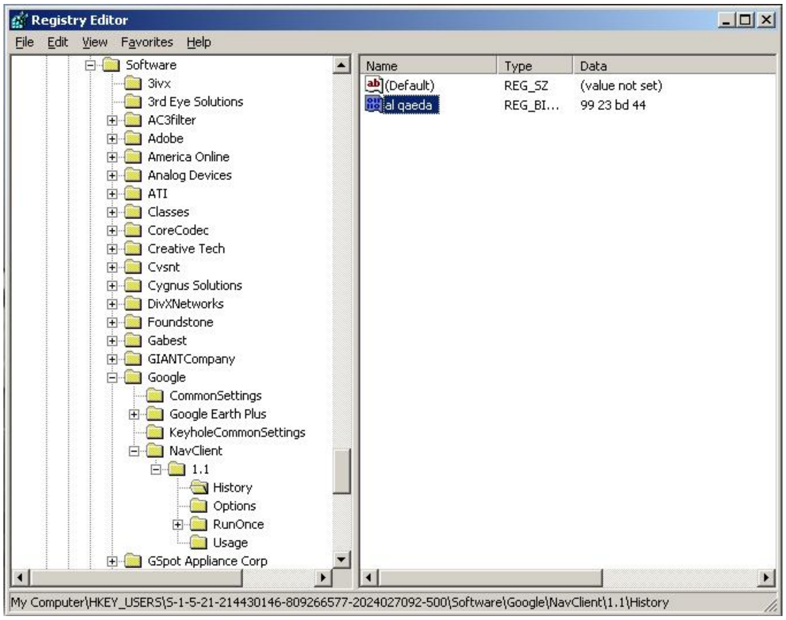


Figure 18 Search for “Al Qaeda”

The search strings in this example, depicted in Figure 19, appear as registry entries within the subkey instead of registry key data due to the way Google Toolbar stores information (e.g. executed searches) in alphabetical order as opposed to storing the data chronologically using placeholders such as 1, 2, 3 or A, B, C. This is better illustrated in the following diagram.

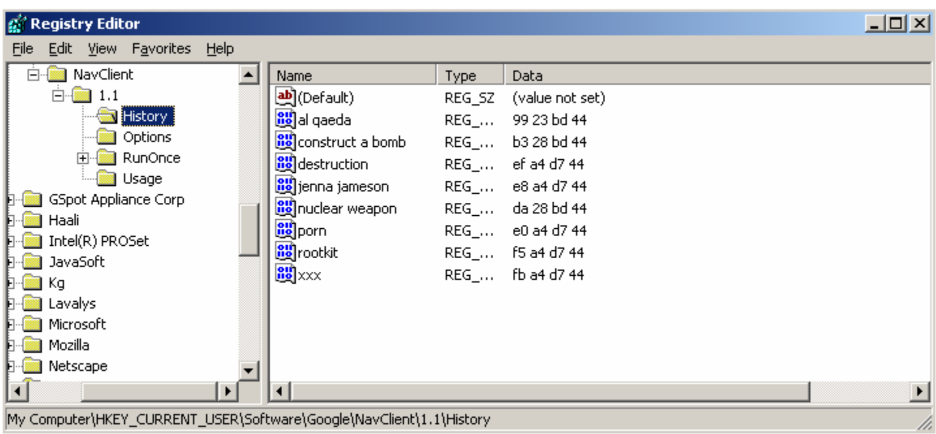


Figure 19 Suspicious Searches

Example 3) User installs a program or new software.

Several registry keys and sub keys are added in the following locations.

HKEY_LOCAL_MACHINE\SOFTWARE

HKEY_CURRENT_USER\Software

HKEY_USERS\.DEFAULT\Software

HKEY_USERS\S-1-5-21-214430146-809266577-2024027092-500\Software where S-1-5-21-214430146-809266577-2024027092-500 is an example of a user identity.

8.2 Analyze Attributes

- Open
- Read
- Move

Example 1) User opens a file named “Al Qaeda Nuclear Program.”

File access appears, as depicted in Figure 20, as a new entry in **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU*** and **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\doc** as well as **HKEY_USERS\S-1-5-21-214430146-809266577-2024027092-500\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU*** and **HKEY_USERS\S-1-5-21-214430146-809266577-2024027092-500\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\doc** where **S-1-5-21-214430146-809266577-2024027092-500** is a unique system identity for a user.

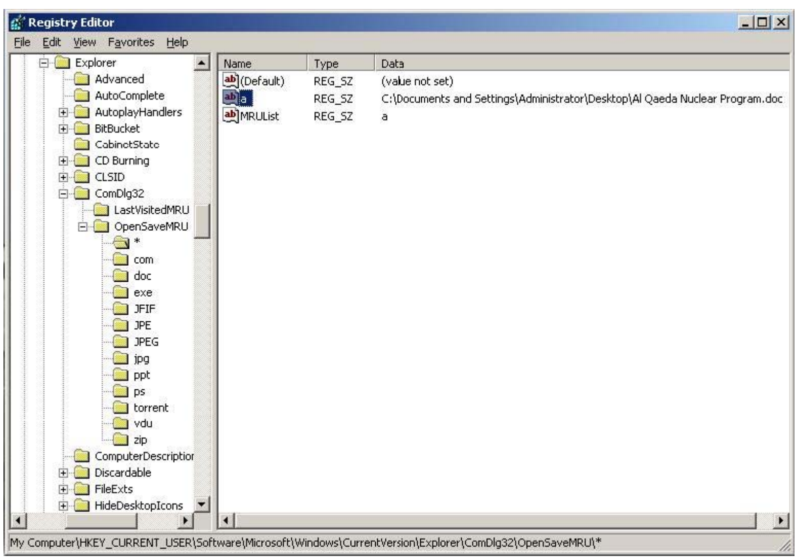


Figure 20 Al Qaeda Nuclear Program File

There are other registry keys and sub keys that indicate programs or services being run, as shown in Figure 21.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServicesOnce

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services

HKEY_CLASSES_ROOT\exefile\shell\open\command

HKEY_CLASSES_ROOT\exefile\shell\runas\command

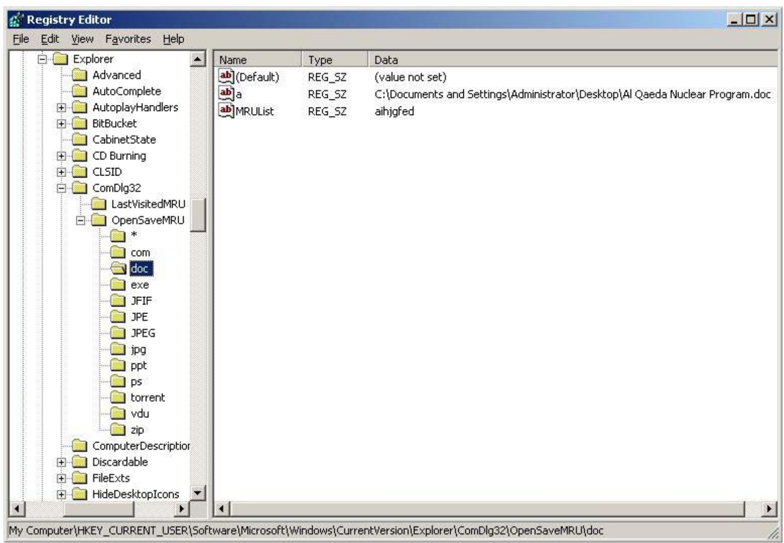


Figure 21 Registry Keys and Sub Keys

Example 2) Access a Microsoft Word Document from one location to another.

This sequence of registry accesses, queries, and closures is the same for moving, cutting, and copying. The name of the file is never observed within the registry. This highlights the important point that there are limitations to applying the Windows Registry to addressing the insider threat. Additional points of data collection are necessary to fully triangulate all possible malicious activities. (Figure 22)

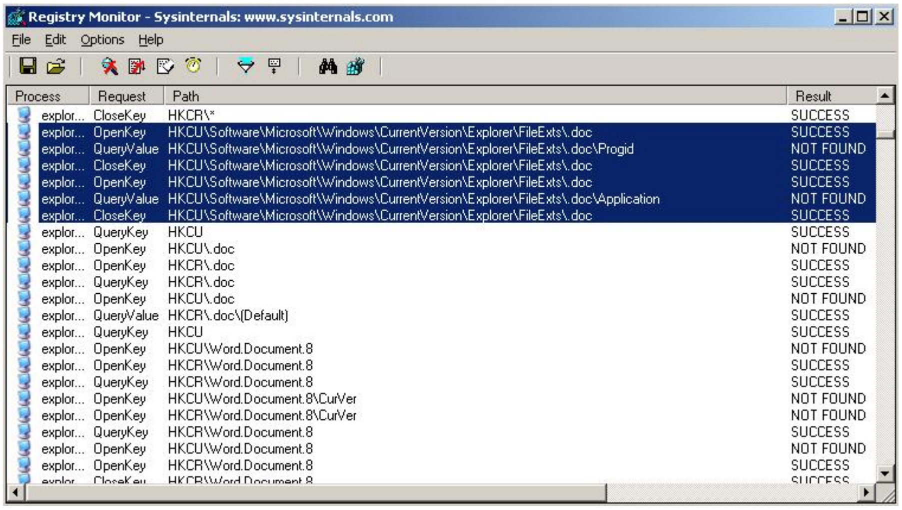


Figure 22 Registry Monitor

Example 3) Access an Adobe Acrobat Document without opening the file.

This sequence of registry accesses, queries, and closures is the same for moving, cutting, and copying. The name of the file is never observed within the registry. The granularity of the data collected and recorded by the Windows Registry is not enough for complete detection of malicious activity. (Figure 23)

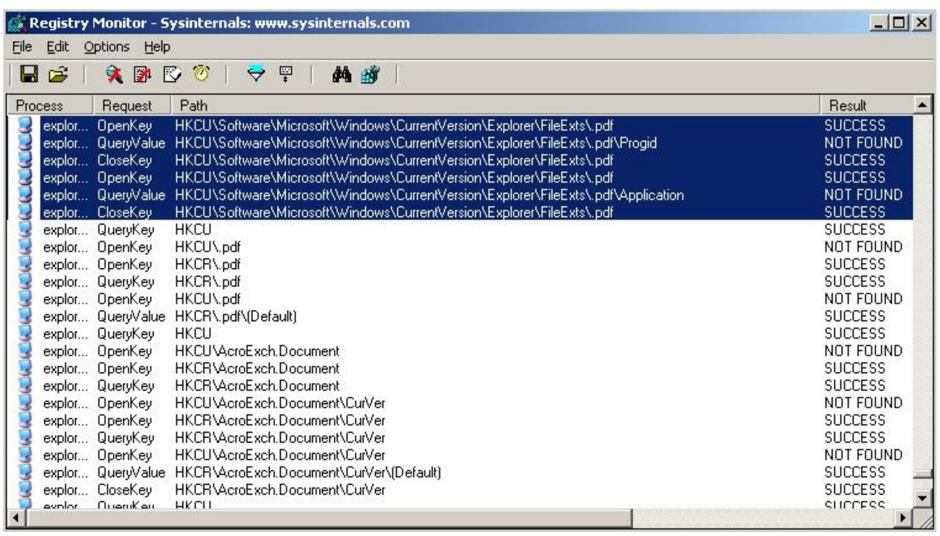


Figure 23 Registry Monitor

8.3 Output Attributes

- Copy
- Create
- Print

Example 1) Save a NOTEPAD text dump out to a Microsoft Word document named “Al Qaeda.doc.” This file is neither open, viewed, or saved from Microsoft Word.

Al Qaeda.doc shows up, depicted in Figure 24, as created in both **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU*** and **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\doc** as well as **HKEY_USERS\S-1-5-21-214430146-809266577-2024027092-500\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU*** and **HKEY_USERS\S-1-5-21-214430146-809266577-2024027092-500\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\doc** where **S-1-5-21-214430146-809266577-2024027092-500** is a unique system identity for a user.

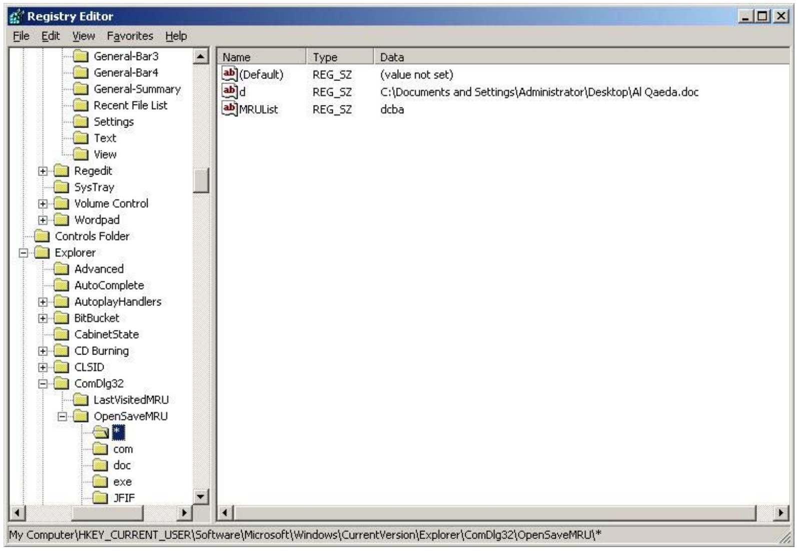


Figure 24 Registry Editor

8.4 Output Attributes

- Copy
- Create
- Print

Example 1) Copying a file or creating a file without using the file handler itself.

The registry activity surrounding this action by the user is identical to a file being moved to a different location, or copied and pasted without using the actual file handler. For instance, if a Microsoft Word Document (.doc) is created (e.g. by renaming), moved to another location, (e.g.

cut and paste) or copied without actually using Microsoft Word to do so, then the registry activity involved is almost identical in every case.

8.5 Transfer Attributes

- Carry
- E-Mail
- FTP
- Telnet
- Publish To WWW

Example 1) Connections made using Telnet or FTP.

HKEY_CURRENT_USER\Software\Microsoft\Telnet and **HKEY_CURRENT_USER\Software\Microsoft\FTP** contains a list of the last ten hosts to which the machine was connected using the respective technologies.

Connections made using Third-Party applications like Putty can also be identified, as shown in Figure 25.

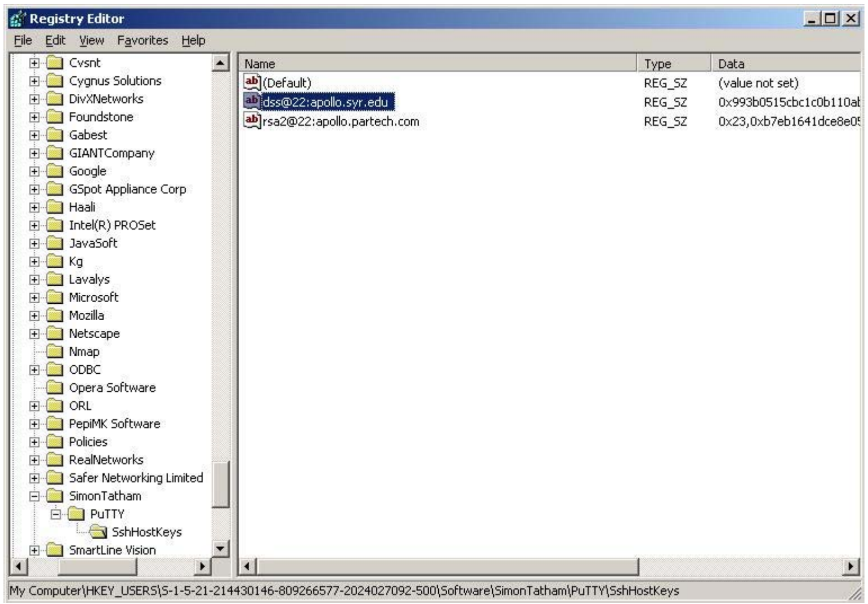


Figure 25 Registry Editor

Registry monitoring of third-party programs and various forms of communication is often sparse, indication another limitation of this approach.

8.6 Limitations of the Registry Approach

There are a number of limitations that exist in using the Windows Registry. The registry pays very little attention to the copying, moving, printing and renaming of files. If a file is opened using a file handler, then that activity gets recorded in the registry. For instance, if a file such as a Microsoft Word Document ("Al Qaeda.doc") gets opened using a file handler such as Microsoft Office's Word program, that activity gets recorded and saved in several places in the registry. These locations include the following.

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU

*****

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU

\doc

HKCU\Software\Microsoft\Windows\ShellNoRoam\MUICache

Unfortunately, if the same Word Document gets selected and acted upon without actually being opened using an associated file handler, then the activity surrounding that file remains unclear to a forensic investigator from the perspective of the registry. In this instance, there will be registry activity that indicates activity surrounding a certain file type, but no definitive evidence is presented as to whether that file was copied, moved, or renamed.

Modification time of different files or actions is not easily accessible via the registry, representing another weakness in this approach. The Operating System records the last time a registry key was changed; however, that information is not easily viewed by an investigator. Loose monitoring of printer activity is another registry weakness that we present here. If a sensitive document is printed by an insider who does not possess the proper authorizations to do so, the registry retains little or no evidence that such an action took place.

There are extensions to the registry approach that must be incorporated in order for this approach to be an all encompassing solution to insider misuse. These extensions include the following.

- Monitoring and recording of the Windows printer queue.
- Monitoring and recording of MAC times for files labeled as critical or sensitive.
- Monitoring for activity that suggests that a file of the same size and type of another is created in the system.

9. WINDOWS REGISTRY VIEW OF ATTACK SCENARIOS

The Microsoft Windows Registry can be used to identify violations of Confidentiality, Integrity, and Availability (CIA) that represent a possible Insider Threat or malicious action, as dicussed in Table 4.

Table 4 Insider Threat CIA

Security Property Violated	Example: Privilege Misuse	Example: Privilege Escalation
Confidentiality	Leaking Sensitive Information	Obtain Ability to Leak Information
Integrity	Changing Security Level of Files	Obtain Ability to Change Integrity
Availability	Perform Denial of Service (DOS)	Obtain Ability to Stop Service

Scenario #1

This scenario involves violating confidentiality through privilege misuse by leaking sensitive information. A malicious insider accesses a sensitive file labeled "X.doc" and changes the filename to "Y.doc" for the purpose of transmitting the file to an unapproved media or an unauthorized party. Transmission could be done using e-mail, printing, copying to external media, as well as any other feasible forms of transmitting digital material.

If Microsoft Word is used for this purpose, then this activity will appear in the following registry keys.

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU*

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\doc

HKCU\Software\Microsoft\Windows\ShellNoRoam\MUICache

HKCU\Software\Microsoft\Office\10.0\Common\Open Find\Microsoft Word\Settings\Save As\File Name MRU

Both the original file "X.doc" and the new file "Y.doc" will appear in the registry. In addition, "Y.doc" will appear to have been created after "X.doc." Also, two files will now exist on the system with the same size and same extension. This situation may be common for system files such as .dll files but is uncommon for two supposedly different documents to be of the same size. This same anomaly appears if a user duplicates a sensitive file under a different name without using the "Save As" function as part of the file handler. In all these instances, the MAC (Modified, Access, Created) times of both the original file and the copy will be changed.

An insider might wish to copy, drag or move a file to a different location or folder. In this case the path of the sensitive file will change, as well as the MAC time. If the file is dragged or moved to an external drive such as a USB thumb drive, then the pointer to the sensitive file will disappear in the Master File Table (MFT) since the file, in this case, will no longer exist on the system.

Scenario #2

This scenario involves violating integrity through privilege misuse by changing the security labels of sensitive information. A malicious insider accesses a Microsoft Word Document ("X.doc") and changes the metadata associated with the file from a security label reflecting sensitive and confidential information to a label that reflects an unclassified security label.

The example below will show that whenever labels or metadata relating to a file are changed, the MAC times will be altered. In Figure 26, all permissions to a sensitive file labeled "X.doc" are denied to users without administrative privilege. Figure 27 shows the MAC times related to "X.doc" with its integrity intact. Figure 28 shows an administrator changing the security properties of the file to allow any user full control and rights to the file. The MAC times of the file change even though the contents have not been opened, saved, or modified in any way.

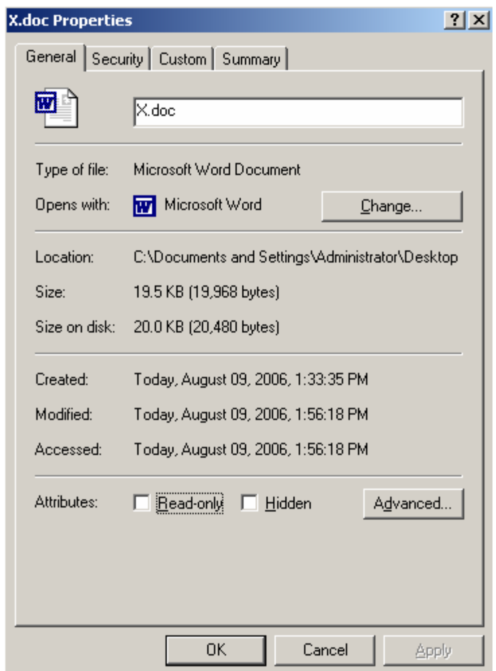


Figure 26 Properties

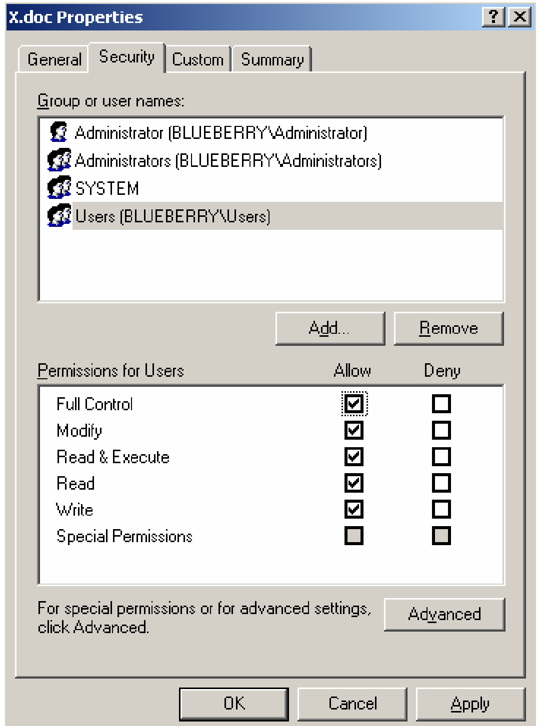


Figure 27 User's Permissions

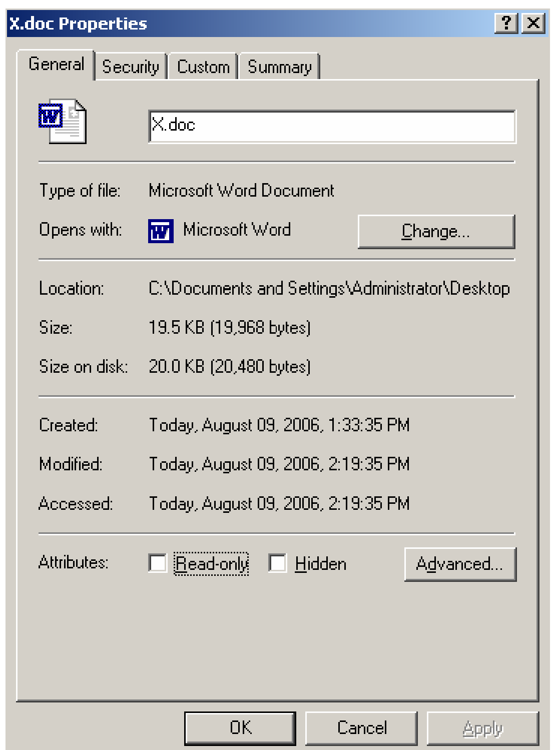


Figure 28 Properties

Scenario 3

This scenario involves violating availability through privilege abuse by performing a Denial of Service (DOS) or stopping a service that should be available to other individuals within the organization. An insider could do considerable damage to an organization by rendering some of its systems or applications useless by a network attack. A system administrator could perform a Denial of Service (DOS) or Distributed (DDOS) on a network machine or resource for the purpose of breaking availability.

DOS related attributes include:

HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

HKLM\SYSTEM\CurrentControlSet\Services

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options

HKLM\System\CurrentControlSet\Control\Terminal Server

HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat

HKLM\System\CurrentControlSet\Control\Terminal Server\TSUserEnabled

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

HKLM\System\CurrentControlSet\Services\WinSock2\Parameters

HKLM\System\CurrentControlSet\Services\DnsCache\Parameters

HKLM\Software\Policies\Microsoft\Windows NT\DnsClient

HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\UseDomainNameDevolution

HKLM\Software\Microsoft\Rpc\PagedBuffers

HKLM\Software\Microsoft\Rpc

HKLM\Software\Microsoft\Rpc\MaxRpcSize

HKLM\Software\Policies\Microsoft\Windows NT\Rpc

HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Hostname

HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Domain

HKLM\Software\Microsoft\Rpc\SecurityService\DefaultAuthLevel

HKLM\Software\Microsoft\Rpc\SecurityService

HKLM\SYSTEM\CurrentControlSet\Services\Winsock\Parameters

HKLM\SYSTEM\CurrentControlSet\Services\Winsock\Parameters\Transports

HKLM\SYSTEM\CurrentControlSet\Services\Winsock\Parameters

HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock

HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\AuthorizedApplications

Other related attributes include:

- Altered HOSTS file.
- Netstat –b will show odd network connections.
- Microsoft Network Monitor will show high network utilization as a result of a propagating worm.
- Nbtstat –a will show odd NETBIOS connections to other machines on the network.

Scenario 4

Malicious code is installed on a system and attempts to evade detection by hiding on the Windows system. This is the technique that rootkits use.

A Rootkit is a malicious software package intended to conceal malicious system processes, files or system data created by the Rootkit, thereby helping an intruder to maintain access to a system whilst avoiding detection by conventional protection tools.

There are many registry attributes known to be utilized by Rootkits due to their complexity. Detecting a Rootkit requires close attention be paid to the registry keys that are responsible for starting programs on the system startup. Those registry keys are listed below.

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\KnownDLLs
HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Session Manager\KnownDLLs
HKEY_LOCAL_MACHINE\System\ControlSet\Services
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\Run
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\RunOnce
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\RunOnceEx
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\WinLogon
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows (run)
HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\Run
HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\RunOnce
HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\RunOnceEx
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows (run)
HKEY_CLASSES_ROOT\exefile\shell\open\command

Installation and uninstallation of applications and services are detected in the Microsoft Windows Event Log, depicted in Figure 29, which can be searched and parsed by the integrated event viewer or several third-party programs, such as Event Log Explorer.

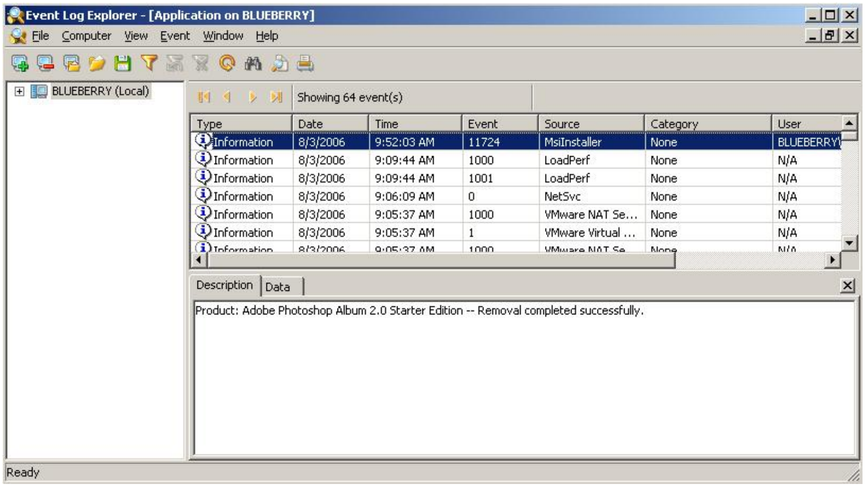


Figure 29 Event Log Explorer

Scenario 5

A malicious insider uses elevated privileges to install and then execute a program called "Cain & Abel" which is used for cracking passwords, SAM files, and conducting other possible harmful functions.

Figure 30 shows a typical response to an installation attempt for a user who does not have the necessary privileges. In these cases the installer would either fail to run or the installation process would encounter unrecoverable errors.

Windows has a built-in function called "RunAs" which allows for a program or executable such as an installation to run with elevated privileges. Figure 31 shows this process taking place. A regular user in this case runs the installation with the credentials of a system administrator. Finally, Figure 32 shows the "Cain & Abel" installation executing without problem under the elevated privileges.

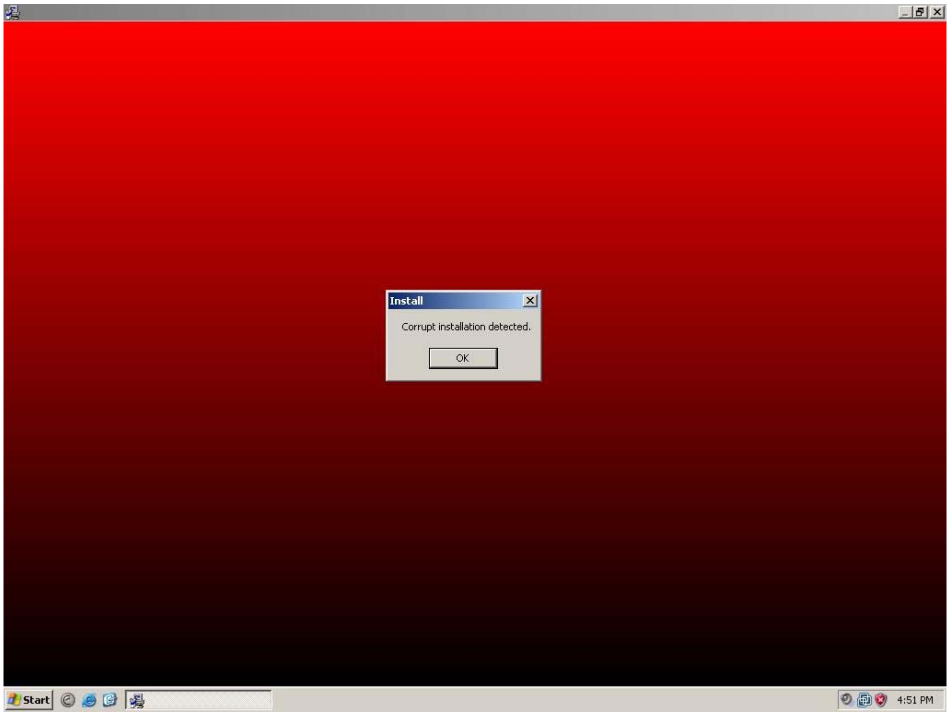


Figure 30 Installation Attempt

The following registry keys are involved in such an activity:

HKCU\exefile\shell\runas

HKCR\exefile\Shell\runas

HKCR\exefile\Shell\runas\LegacyDisable

HKCR\exefile\Shell\runas\CheckSupportedTypes

HKCR\exefile\Shell\open

HKCR\exefile\shell\open\command\(Default)

HKLM\SECURITY\Policy\SecDesc\(Default)

HKLM\SECURITY\Policy\SecDesc

HKLM\SAM\SAM\DOMAINS\Builtin\Groups

HKLM\SAM\SAM\DOMAINS\Builtin\Aliases

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU*

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\exe

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

HKCU\Software\Microsoft\Windows\ShellNoRoam\MUICache

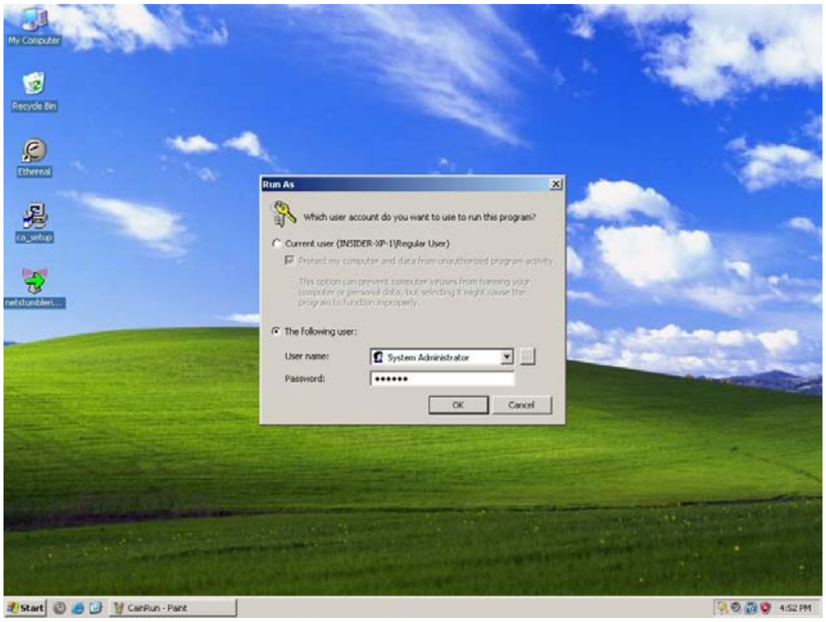


Figure 31 Run As

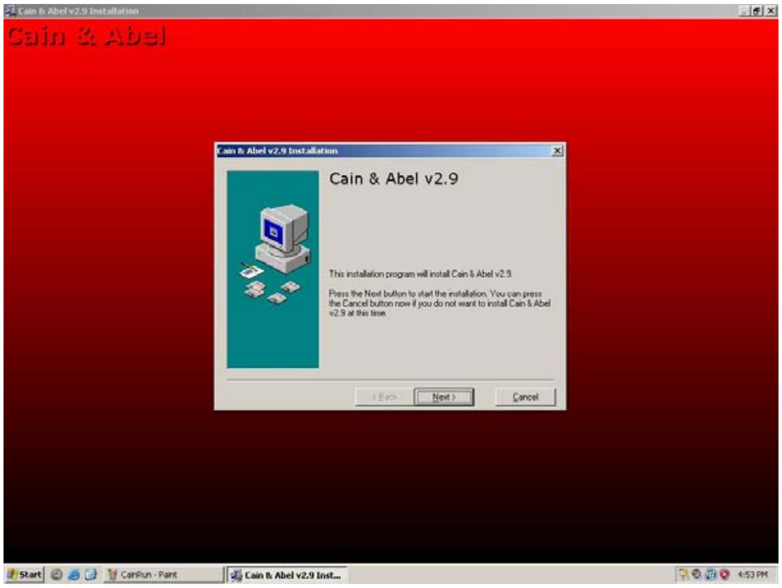


Figure 32 Installation

10. ADDRESSING INSIDER THREATS WITH FILEMON

Given a sensitive file ("File X") which for our purposes in this demonstration will be a Microsoft Word Document called "X.doc," how can an investigator detect in real-time the following three scenarios?

1. Detect renaming "File X" to "File Y," such as renaming "X.doc" to "Y.doc." "X.doc" is a sensitive file while there is no classification or protection regarding "Y.doc."
2. Detect copying "File X." This is a copy and paste operation.
3. Detect moving "File X" from current working directory to another. This is either a cut and paste or copy and move operation.

Scenario 1

User attempts to rename file "X.doc" to "Y.doc" are shown in the Figures 34 and 35, below. This could possibly indicate that an insider is trying to move a sensitive file under the guise of a non-sensitive file. Figure 33 depicts the user opening Filemon.

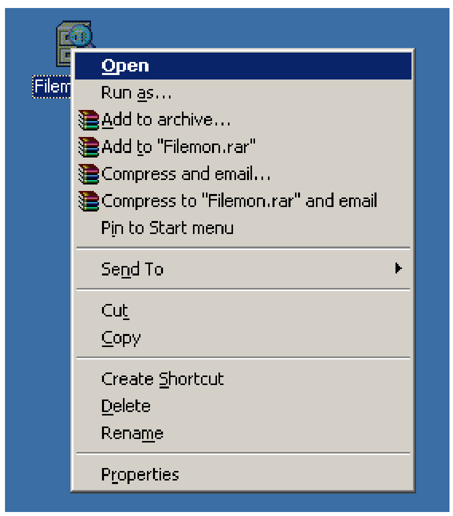
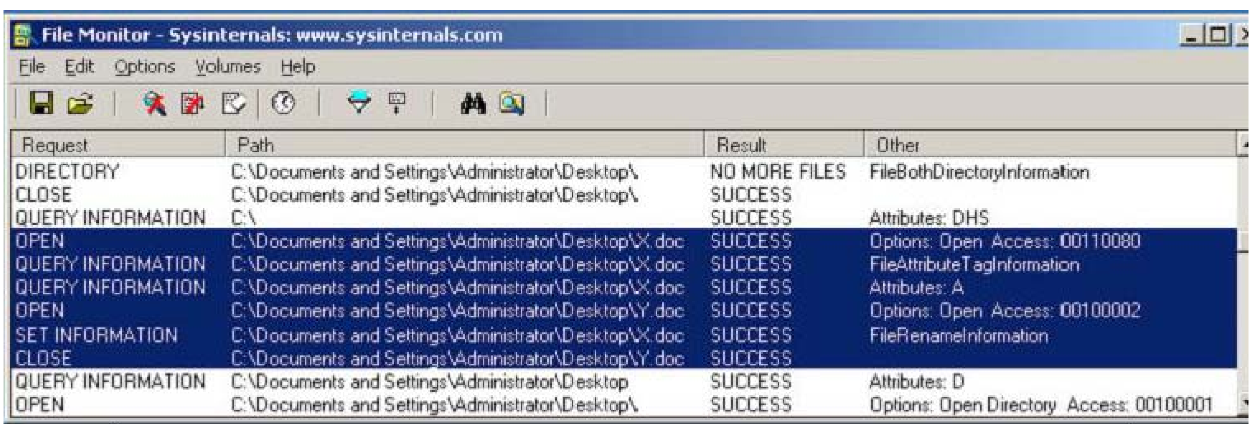


Figure 33 Opening Filemon

Figure 34 Rename

Figure 35 New Name

Filemon very explicitly captures and identifies this action as shown in Figure 36. It is observable that the system checks the files attributes and then performs a file rename. In the midst of all this activity the system also creates the container file "Y.doc."



Request	Path	Result	Other
DIRECTORY	C:\Documents and Settings\Administrator\Desktop\	NO MORE FILES	FileBothDirectoryInformation
CLOSE	C:\Documents and Settings\Administrator\Desktop\	SUCCESS	
QUERY INFORMATION	C:\	SUCCESS	Attributes: DHS
OPEN	C:\Documents and Settings\Administrator\Desktop\X.doc	SUCCESS	Options: Open Access: 00110080
QUERY INFORMATION	C:\Documents and Settings\Administrator\Desktop\X.doc	SUCCESS	FileAttributeTagInformation
QUERY INFORMATION	C:\Documents and Settings\Administrator\Desktop\X.doc	SUCCESS	Attributes: A
OPEN	C:\Documents and Settings\Administrator\Desktop\Y.doc	SUCCESS	Options: Open Access: 00100002
SET INFORMATION	C:\Documents and Settings\Administrator\Desktop\X.doc	SUCCESS	FileRenameInformation
CLOSE	C:\Documents and Settings\Administrator\Desktop\Y.doc	SUCCESS	
QUERY INFORMATION	C:\Documents and Settings\Administrator\Desktop	SUCCESS	Attributes: D
OPEN	C:\Documents and Settings\Administrator\Desktop\	SUCCESS	Options: Open Directory Access: 00100001

Figure 36 File Monitor

Scenario 2

An insider might wish to copy a sensitive file, as depicted in Figures 37 and 38, for the purpose of transmission or carry. Since the original file remains in its proper location the insider might avoid suspicion.

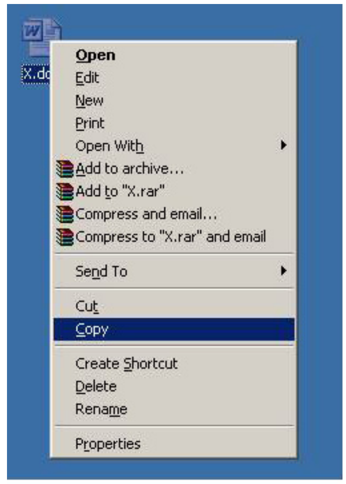


Figure 37 Copy Function

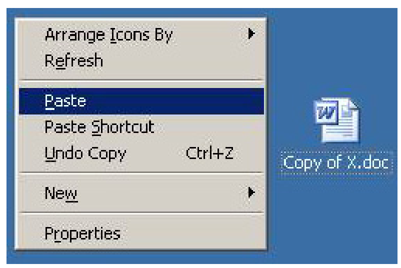


Figure 38 Paste Function

Filemon captures and identifies this activity by detecting a name collision, as depicted in Figure 39. If a user tries to copy a file to the same directory that the file is already residing in then there is a name collision and the new file has to be given a different name, in this case "Copy of X.doc." The process of checking to see if "X.doc" and "Copy of X.doc" exist, then copying "X.doc" and creating "Copy of X.doc" is shown below.

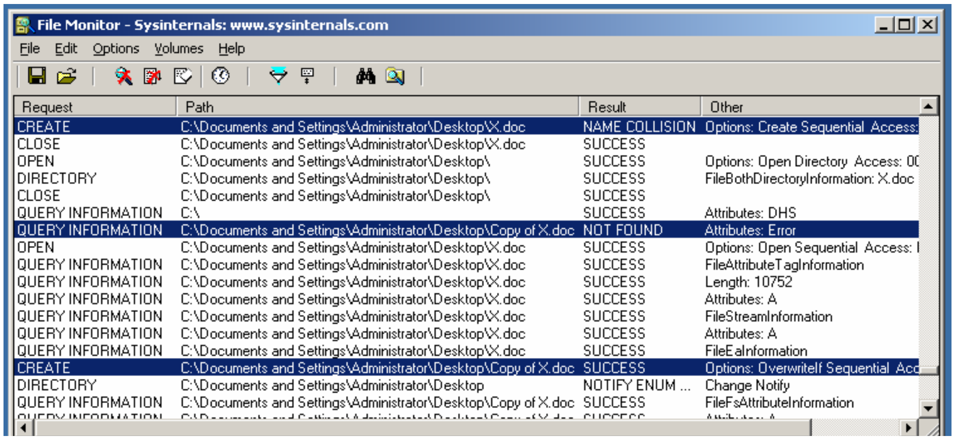


Figure 39 File Monitor

Scenario 3

An insider might sneak an external drive such as a USB thumb drive into a classified facility in order to carry sensitive files. The insider must transfer (move) the desired sensitive file onto the external drive, which for this demo we will represent as another directory, depicted in Figures 40 through 42.

Figure 40 External Drive Folder

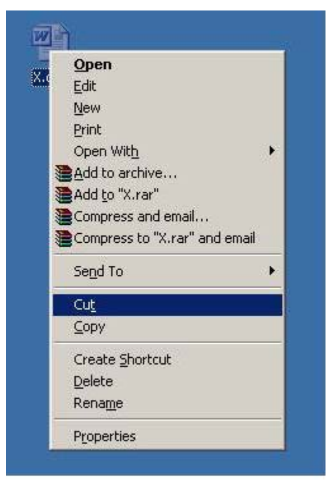


Figure 41 Cut Function

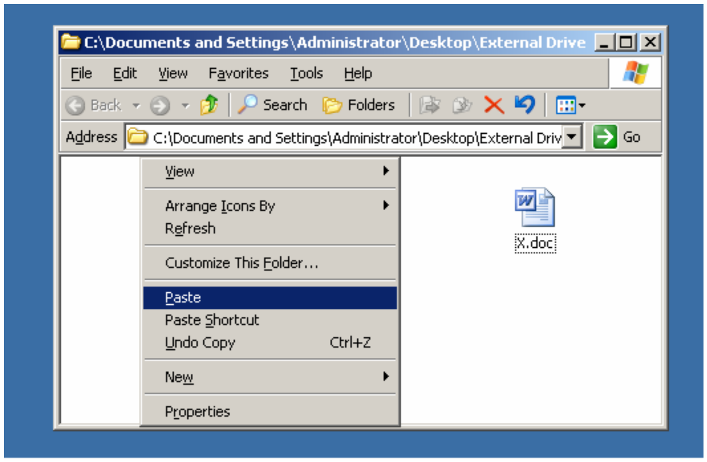


Figure 42 Paste Function

Filemon detects and identifies this activity on the system by registering a file rename. Since "X.doc" is moved to another location then in essence, it must be renamed since the path (physical location) is changing. The process of "X.doc" being renamed and moved to the directory "External Drive" is depicted in Figure 43.

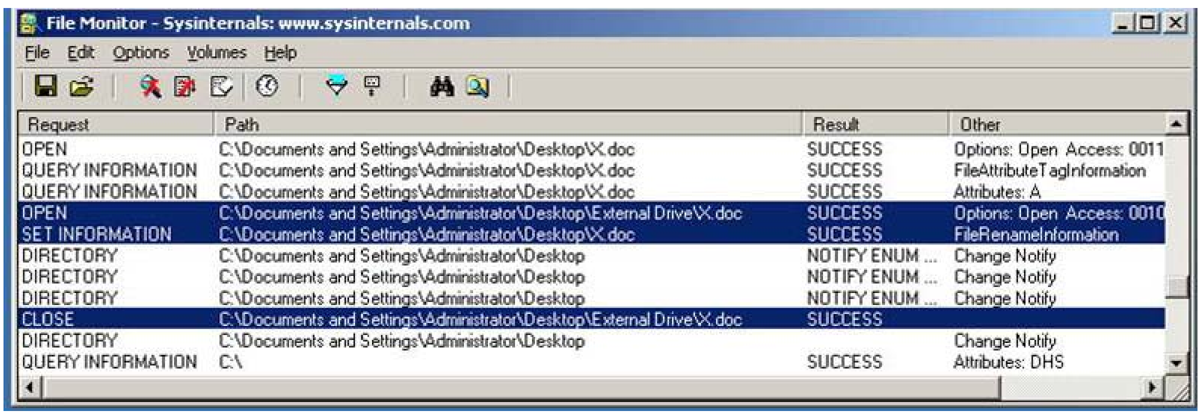


Figure 43 File Monitor

APPENDIX A: ACRONYM LIST

Acronym	Definition
AAC	Active Access Control
ACE	Advanced Course in Engineering
ACL	Access Control List
ADS	Active Directory Services
AFOSR	Air Force Office of Scientific Research
AFRL/IF	Air Force Research Laboratory/Information Directorate
API	Application Programming Interface
CLI	Command Line Interface
CRS	Central Role Server
DAC	Discretionary Access Control
DBX	Digital Private Branch Exchange
DLL	Data Link Layer
DoD	Department Of Defense
DOS	Denial Of Service
ELF	Executable and Linkable Format
FASAC	Fine-Grained, Active, and Scalable Access Control
FAST	Forensic Analyst’s Software Tool
FTP	File Transfer Protocol
GMT	Global Media Transfer
HKCC	HKEY_CURRENT_CONFIG
HKCR	HKEY_CLASSES_ROOT
HKCU	HKEY_CURRENT_USER
IDE	Interactive Development Environment
IDS	Intrusion Detection System
IE	Internet Explorer
IPv6	Internet Protocol version 6

IRC	Internet Relay Chat
IS	Information System
IT	Information Technology
MAC	Mandatory Access Control
MFT	Master File Table
OS	Operating System
PAR	PAR Government Systems Corporation
PI	Principal Investigator
PRA	Permission Role Assignment
RBAC	Role-Based Access Control
SAM	Security Accounts Manager
SMTP	Simple Mail Transfer Protocol
TCP/IP	Transfer Control Protocol/Internet Protocol
TIF	Image Format
TSA	Transportation Security Administration
UDP	User Datagram Protocol
URA	User Role Assignment
USB	Universal Serial Bus
VoIP	Voice over Internet Protocol
WWW	World Wide Web

APPENDIX B: REFERENCES

[AS00] Gail-Joon Ahn and Ravi S. Sandhu. *Role-based Authorization Constraints Specification*. ACM Transactions on Information and System Security, 3(4):207-226, November 2000.

[BA04] Richard C. Brackney, Robert H. Anderson. *Understanding the Insider Threat*. In Proceedings of the March 2004 Workshop (Prepared for the Advanced Research and Development Activity), March 2004.

[CPSGB02] Vicka Corey, Charles Peterman, Sybil Shearin, Michael S. Greenberg, James Van Bokkelen. *Network Forensics Analysis*. IEEE Internet Computing, Volume 6, Issue 6, pages 60-66, November 2002.

[FSGetal01] David F. Ferraiolo, Ravi S. Sandhu, Serban Gavrila, D. Richard Kuhn, and RamaSwamy Chandramouli. Proposed NIST standard for role-based access control. *ACM Transactions on Information and Systems Security*, 4(3):224-274, August 2001.

[Gar01] Lee Garber. Computer *Forensics: High-Tech Law Enforcement*. IEEE Computer, Volume 34, Issue 1, pages 22-27, January 2001.

[LM02] Ninghui Li, John C. Mitchell, and William H. Winsborough. Design of a role-based trust management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 114-130. IEEE Computer Society Press, May 2002.

[LT04] Ninghui Li and Mahesh V. Tripunitara. Security analysis in role-based access control. In *Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies (SACMAT 2004)*, June 2004.

[Neu99] Peter G. Neumann. *Inside Risks: Risks of Insiders*. Communications of the ACM, Volume 42 Issue 12, December 1999.

[NIST02] National Institute of Standards and Technology. The economic impact of role-based access control. Planning Report 02-1, March 2002. available at http://www.nist.gov/director/progofc/ reportO2l.pdf.

[OR03] Rolf Oppliger, Ruedi Rytz. *Digital Evidence: Dream and Reality*. IEEE Security and Privacy, Volume 1, Number 5, pages 44-48, September 2003.

[PH04] Joon S. Park and Shuyuan M. Ho. *Composite Role-Based Monitoring (CRBM) for Countering Insider Threats*. Proceedings of the 2nd Symposium on Intelligence and Security Informatics, Tucson, Arizona, June 10-11, 2004.

[PS00] Joon S. Park and Ravi Sandhu. *Binding Identities and Attributes Using Digitally Signed Certificates*. 16th Annual Computer Security Applications Conference (ACSAC), New Orleans, Louisiana, December 11-15, 2000.

[PS99b] Joon S. Park and Ravi Sandhu. *RBAC on the Web by Smart Certificates*. 4th ACM Workshop on Role-Based Access Control, ACM, Fairfax, Virginia, October 28-29, 1999.

[PSA01] Joon S. Park, Ravi Sandhu, and Gail-Joon Ahn. *Role-based Access Control on the Web*. ACM Transactions on Information and System Security (TISSEC), 4(1), February 2001.

[PSU04] Suranjan Pramanik, Vidyaraman Sankaranarayanan, Shambhu Upadhyaya. *Security Policies to Mitigate Insider Threat in the Document Control Domain*. In Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC04), pages 304-313, Tucson, Arizona, December 2004.

[RJ05] Wei Ren, Hai Jin. Distributed Agent-Based Real Time Network Intrusion Forensics System Architecture Design. In Proceedings of the19th International Conference on Advanced Information Networking and Applications (AINA'05), Volume 1 (AINA papers), pages 177-182, March 2005.

[SCFY96] R. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. *Role Based Access Control Models*. *IEEE Computer 29 (2)*, February 1996.

Frank Apap, Andrew Honig, Shlomo Hershkop, Eleazar Eskin, Sal Stolfo. *Detecting Malicious Software by Monitoring Anomalous Windows Registry Accesses.* CiteSeer. 2002.

Robbert van Renesse, Kenneth P. Birman, Werner Vogels. *Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring, Management, and Data Mining.* CiteSeer. 2001.

Greg Shultz. *The Anatomy of the Windows Registry.* TechRepublic. 2005. http://techrepublic.com.com/i/tr/downloads/home/anatomy_of_windows_registry.pdf

Mike Lewis. *Understanding the Registry*. PC Support Advisor. 1999. Pages 5-10.

Jerry Honeycutt. *Microsoft Windows XP Registry Guide.* Microsoft Press. 2002.

Eric Cole, Sandra Ring. *Insider Threat*. Syngress Media Inc. 2006.

Anthony Steven. *The Security Monitoring and Attack Prevention Guide*. Microsoft. 2005.

Katherine Heller, Krysta Svore, Angelos Keromytis, Salvatore Stolfo. *OCSVM for Detecting Anomalous Windows Registry Accesses*. CiteSeer. 2002.

Vic Ferri. *Registry Data Types*. TechTrax.
http://pubs.logicalexpressions.com/Pub0009/LPMArticle.asp?ID=361

Thornton, John I. (1997), "The General Assumptions And Rationale Of Forensic Identification", in David L. Faigman, David H. Kaye, Michael J. Saks, & Joseph Sanders, *Modern Scientific Evidence: The Law And Science Of Expert Testimony*, vol. 2, St. Paul: West Publishing Co.

Wong, Li Wern. *Forensic Analysis of the Windows Registry*. Forensic Focus. 2005.
http://www.forensicfocus.com/downloads/forensic-analysis-windows-registry.pdf

APPENDIX C: ADDITIONAL DIAGRAMS

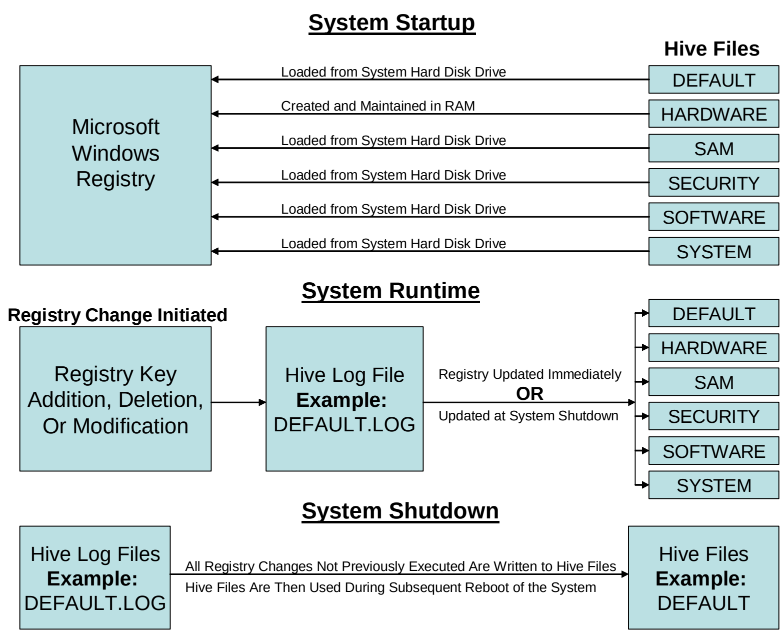


Figure 44 Microsoft Registry Operation

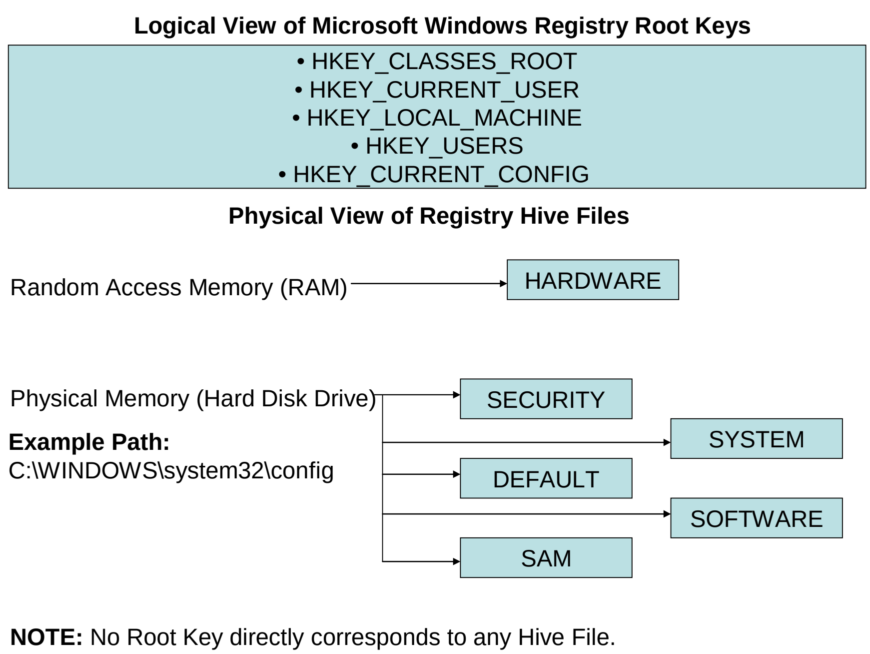


Figure 45 Logical View of the Windows Registry

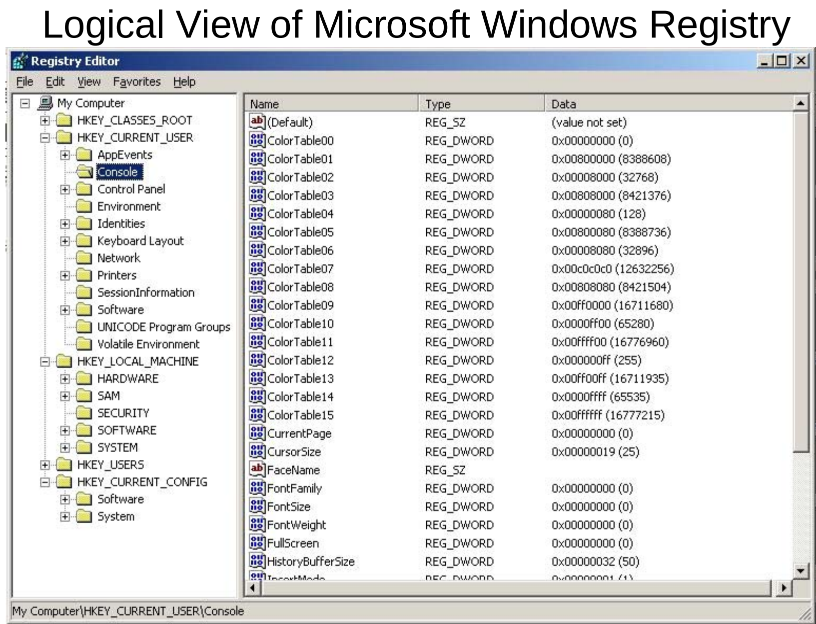
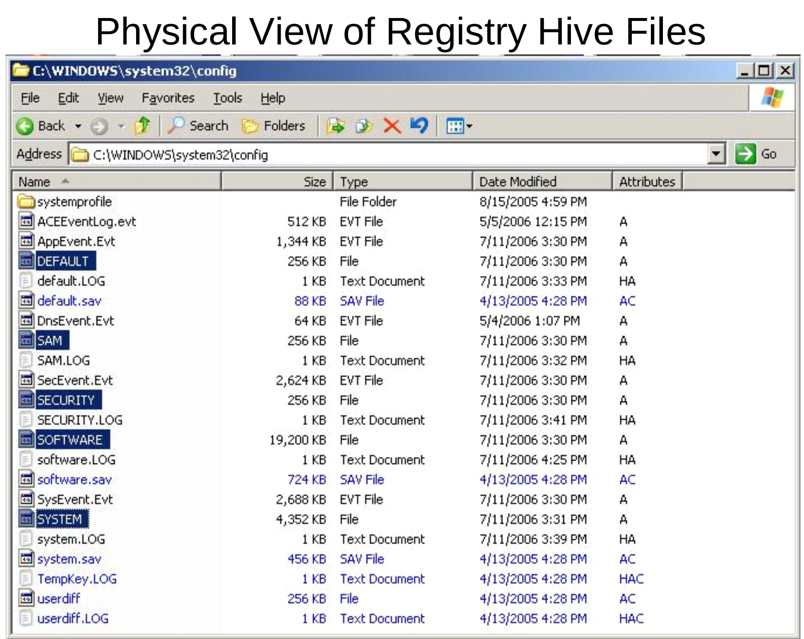


Figure 46 Registry View



NOTE: HARDWARE Hive is stored completely in Random Access Memory.

Figure 47 Hive Files

APPENDIX D: ADDITIONAL SCENARIOS

The scenarios below demonstrate the different state changes recognized by the Windows Operating System when possible malicious activity is initiated by either a regular user, privileged user such as a system administrator, or an outsider threat.

Regular User – This entity is an insider without administrator or root access on most system and one who might have to elevate their privilege in order to abuse the system.

System Administrator – This entity is an insider who already has the necessary privileges on the system and uses those privileges to abuse the system.

Outsider – This entity is an external threat that must subvert a system internal to the network or leverage an insider in order to abuse the system.

Table 5 Additional Scenarios

Threat	Breaking Confidentiality / Integrity	Breaking Availability
Regular User	Leaks information using E-Mail, FTP, Telnet, or the WWW.	Deletes documents or files belonging to the organization, either on the local machine or network shares.
System Administrator	Spreads malicious code using already acquired system capabilities.	Performs a Denial of Service (DOS) or Distributed (DOS) on a network machine or resource.
Outsider	Installs a Rootkit on an insider machine.	Compromises a network system for purpose of creating a zombie, Botnet, or SPAM box.

D.1 Scenario #1

A malicious insider could become disgruntled and try to expose trade secrets or sensitive information regarding the organization. A regular user could leak information using E-mail, FTP, Telnet, or the WWW for the purpose of breaking confidentiality or integrity. In this instance, an insider could be embedding sensitive information within E-mail attachments sent to outside entities, transferring files to outside computers using FTP or Telnet, or publishing information to the public using WWW forums or even domains such as MySpace.com.

The Windows Registry can detect activity using Microsoft Outlook, Outlook Express, as well as the FTP and Telnet programs installed with Microsoft Windows. The registry has also been shown to reflect activity in third-party networking applications such as the popular Putty program.

E-Mail attributes:

HKCU\Software\Microsoft\Internet Account Manager\Accounts

HKCU\Software\Microsoft\Windows\CurrentVersion\UnreadMail

HKCR\LDAP\shell\open\command

HKCR\mailto\shell\open\command

HKCR\Microsoft Internet Mail Message\shell\open\command

HKLM\SOFTWARE\Classes\LDAP\shell\open\command

HKLM\SOFTWARE\Classes\mailto\shell\open\command

HKLM\SOFTWARE\Classes\Microsoft Internet Mail Message\shell\open\command

E-Mail attachments:

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Personal

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders\Personal

Telnet attributes:

HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths\telnet.exe

HKCU\Applications\telnet.exe

HKCR\Applications\telnet.exe

HKCU\Software\Microsoft\Windows\ShellNoRoam\MUICache\C:\WINDOWS\system32\telnet.exe

HKLM\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Custom\telnet.exe

HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\telnet.exe

HKLM\System\CurrentControlSet\Control\Terminal Server

HKCU\Software\Microsoft\Telnet

HKLM\SOFTWARE\Microsoft\TelnetServer

FTP attributes:

HKCR\ftp

HKCU\Software\Microsoft\FTP

HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProtocolDefaults

HKLM\SOFTWARE\Classes\ftp

HKLM\SOFTWARE\Microsoft\Internet Explorer\AdvancedOptions\BROWSE\FTPUI

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellCompatibility\Applications\cutftp32.exe

D.2 Scenario #2

For the purpose of financial gain, corporate espionage, or other malicious intent an insider might wish to alter, remove, or adulterate sensitive files. A regular user could delete documents or files belonging to the organization, either on the local machine or network shares for the purpose of breaking availability.

The registry attributes below are associated with selecting a file without opening it. Using the Windows Registry alone it is tough to determine exactly what action has taken place on a file besides opening or saving. Determining a pattern of events that suggests file deletion will most likely take differential and statistical analysis of several data sets (system state captures over time).

HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts

HKCU\SystemFileAssociations

HKCU*\ShellEx\DataHandler

HKCR*\ShellEx\DataHandler

HKCU\AllFilesystemObjects

HKCR\AllFilesystemObjects

HKCU*\shellex\ContextMenuHandlers

HKCR*\shellex\ContextMenuHandlers

HKCU*\shellex\ContextMenuHandlers\Offline Files

HKCR*\shellex\ContextMenuHandlers\Offline Files

HKCR*\shellex\ContextMenuHandlers\Offline Files\SuppressionPolicy

HKCU*\shellex\ContextMenuHandlers\Open With

HKCR*\shellex\ContextMenuHandlers\Open With

HKCR*\shellex\ContextMenuHandlers\Open With\SuppressionPolicy

HKCU*\shellex\ContextMenuHandlers\Open With EncryptionMenu

HKCR*\shellex\ContextMenuHandlers\Open With EncryptionMenu

HKCU\AllFilesystemObjects\shellex\ContextMenuHandlers\Send To

HKCR\AllFilesystemObjects\shellex\ContextMenuHandlers\Send To

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellCompatibility\Objects

HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths

HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer

HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer

HKCU\Software\Microsoft\Windows\ShellNoRoam\MUICache

HKLM\SECURITY\Policy

HKLM\SECURITY\Policy\SecDesc

The following registry attributes are associated with emptying the Recycling Bin.

HKCR\Folder

HKCU\Folder\Shell

HKCR\Folder\Shell

HKCU\Folder\shell\open

HKCR\Folder\Shell\open

HKCU\Folder\shell\explore

HKCR\Folder\Shell\explore

HKCU\AppEvents\Schemes\Apps\.Default\MenuPopup\.Current

HKCU\AppEvents\Schemes\Apps\Explorer\EmptyRecycleBin\.Current

Other related attributes include:

- Nbtstat –a will show odd NETBIOS connections to other machines on the network.

NBTSTAT [[-a RemoteName] [-A IP address] [-c] [-r] [-R] [-RR] [-s] [-S] [interval]]

Local Area Connection:
Node IP Address: [192.168.130.81] Scope Id: []

NetBIOS Remote Cache Name Table

Name	Type	Host Address	Life [sec]	
XRX8D9BE9	<00>	UNIQUE	192.168.130.192	155

Local Area Connection 2:
Node IP Address: [192.168.130.89] Scope Id: []

NetBIOS Remote Cache Name Table

Name	Type	Host Address	Life [sec]	
XRX8D9BE9	<00>	UNIQUE	192.168.130.192	155
LONGM	<20>	UNIQUE	192.168.130.33	380

D.3 Scenario #3

An insider who has become disgruntled could wish to damage network systems to get back at those who he feels has wronged him. A System administrator could spread malicious code using already acquired system capabilities for the purpose of breaking confidentiality or integrity.

Malware can be spread easily by an insider either purposely or on accident. Once a worm gets launched on a network within the perimeter defenses it can become a real inconvenience. The registry keys most often associated with running malware are listed below.

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\KnownDLLs

HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Session Manager\KnownDLLs

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\Run

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\RunOnce

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\RunOnceEx

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows ("run=" line)

HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\Run

HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\RunOnce

HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\RunOnceEx

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices

HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows ("run=" value)

Other related attributes include:

- Microsoft Network Monitor will show high network utilization.
- Netstat –b will show odd network connections.

Windows NETSTAT displays protocol statistics and current TCP/IP network connections.

NETSTAT [-a] [-b] [-e] [-n] [-o] [-p proto] [-r] [-s] [-v] [interval]

-a Displays all connections and listening ports.

-b Displays the executable involved in creating each connection or listening port. In some cases well-known executables host multiple independent components, and in these cases the sequence of components involved in creating the connection or listening port is displayed. In this case the executable name is in [] at the bottom, on top is the component it called, and so forth until TCP/IP was reached. Note that this option can be time consuming and will fail unless the user has sufficient permissions.

-e Displays Ethernet statistics. This may be combined with the –s option.

-n Displays addresses and port numbers in numerical form.

-o Displays the owning process ID associated with each connection.

-p Shows connections for the protocol specified by protocol; protocol may be any of: TCP, UDP, TCPv6, or UDPv6. If used with the –s option to display per-protocol statistics, proto may be any of: IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.

-r Displays the routing table.

-s Displays per-protocol statistics. By default, statistics are shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6; the -p option may be used to specify a subset of the default.

-v When used in conjunction with -b, will display sequence of components involved in creating the connection or listening port for all executables.

D.4 Scenario #4

An insider could do considerable damage to an organization by rendering some of its systems or applications useless by a network attack. A System administrator could perform a DOS or Distributed DOS on a network machine or resource for the purpose of breaking availability.

DOS related attributes include:

HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

HKLM\SYSTEM\CurrentControlSet\Services

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options

HKLM\System\CurrentControlSet\Control\Terminal Server

HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat

HKLM\System\CurrentControlSet\Control\Terminal Server\TSUserEnabled

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

HKLM\System\CurrentControlSet\Services\WinSock2\Parameters

HKLM\System\CurrentControlSet\Services\DnsCache\Parameters

HKLM\Software\Policies\Microsoft\Windows NT\DnsClient

HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\UseDomainNameDevolution

HKLM\Software\Microsoft\Rpc\PagedBuffers

HKLM\Software\Microsoft\Rpc

HKLM\Software\Microsoft\Rpc\MaxRpcSize

HKLM\Software\Policies\Microsoft\Windows NT\Rpc

HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Hostname

HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Domain

HKLM\Software\Microsoft\Rpc\SecurityService\DefaultAuthLevel

HKLM\Software\Microsoft\Rpc\SecurityService

HKLM\SYSTEM\CurrentControlSet\Services\Winsock\Parameters

HKLM\SYSTEM\CurrentControlSet\Services\Winsock\Parameters\Transports

HKLM\SYSTEM\CurrentControlSet\Services\Winsock\Parameters

HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Winsock

HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\AuthorizedApplications

Other related attributes include:

- Altered HOSTS file.
- Netstat –b will show odd network connections.
- Microsoft Network Monitor will show high network utilization as a result of a propagating worm.
- Nbtstat –a will show odd NETBIOS connections to other machines on the network.

D.5 Scenario #5

An outsider installs a Rootkit on an insider machine for the purpose of breaking confidentiality or integrity.

A Rootkit is a malicious software package intended to conceal malicious system processes, files or system data created by the Rootkit thereby helping an intruder to maintain access to a system whilst avoiding detection by conventional protection tools.

There are many registry attributes utilized by Rootkits due to their complexity. Detecting a Rootkit requires that close attention be paid to the registry keys that are responsible for starting programs on the system startup. Those registry keys are listed below.

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\KnownDLLs
HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Session Manager\KnownDLLs
HKEY_LOCAL_MACHINE\System\ControlSet\Services
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\Run
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\RunOnce
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current Version\RunOnceEx
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\WinLogon

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows (run)
HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\Run
HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\RunOnce
HKEY_CURRENT_USER\Software\Microsoft\Windows\Current Version\RunOnceEx
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows (run)
HKEY_CLASSES_ROOT\exefile\shell\open\command

Installation and uninstallation of applications and services are detected in the Microsoft Windows Event Log, depicted in Figure 48, which can easily be searched and parsed by the integrated event viewer or several third-party programs, such as Event Log Explorer.

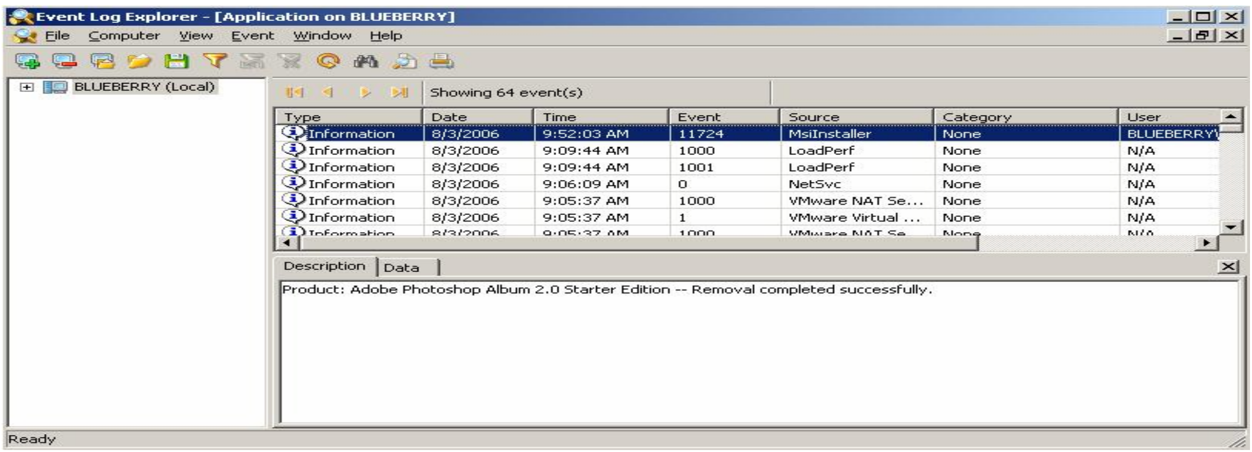


Figure 48 Event Log

D.6 Scenario #6

An outsider compromises a network system for the purpose of creating a zombie, Botnet, or SPAM box to break availability.

Zombie – a system that has been compromised, usually unbeknownst to the system operator, for the purpose of performing malicious actions against other machines or operating as part of a botnet.

Botnet – a collection of compromised machines running malicious software programs such as worms, backdoors, or rootkits that are all under the control of a common perpetrator. A botnet can be controlled remotely through means such as Internet Relay Chat (IRC – RFC 1459).

SPAM – unsolicited, bulk E-mail messages usually perpetrated by abusing SMTP servers and network domains.

The following registry attributes are commonly manipulated and accessed when another entity has control of a victim system.

HKLM\SYSTEM\CurrentControlSet\Services\Remote Administration Service

HKLM\SYSTEM\CurrentControlSet\Services\Remote Administration Service\Security

HLKM\SYSTEM\CurrentControlSet\Services\Remote Administration Service\Enum

HKLM\SYSTEM\ControlSet001\Services\Remote Administration Service

HKCR\exefile\shell\open\command

HKLM\Software\Microsoft\Windows\Current Version\Run

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices

Other related attributes that get modified are:

- Modified windll.dll dynamic link library file.
- Altered HOSTS file.
- Netstat –b will show odd network connections.
- Microsoft Network Monitor will show high network utilization as a result from Botnet communications.
- FPort by Foundstone will identify executables using uncommon protocols such as IRC.

FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

PID	Process		Port	Protocol	Path
1008	DCOM	->	135	TCP	
928	RDP	->	3389	TCP	
816	IEXPLORE	->	1035	TCP	C:\Program Files\Internet Explorer\IEXPLORE.EXE
816	IEXPLORE	->	1036	TCP	C:\Program Files\Internet Explorer\IEXPLORE.EXE
816	IEXPLORE	->	1037	TCP	C:\Program Files\Internet Explorer\IEXPLORE.EXE
1136	NNTP	->	123	UDP	
1008	SMB	->	445	UDP	
4	System	->	4500	UDP	
4	System	->	500	UDP	

D.7 Supplemental Scenarios

D.7.1 Supplemental Scenario 1 – Misuse by System Administrator

The user downloads, installs and runs a sniffing utility. (e.g. Ethereal; however, this could be any type or kind of malicious software)

The domain name of the website where the malicious software is downloaded from will immediately show up in the registry as long as the address was typed in using the address bar in a web browser, as shown below. The ethereal.com domain shows up in a list of other sights that have been visited by this particular Windows system in the recent past. The Windows Registry can be used to track malicious or questionable websites that are being visited by a particular user.

Figure 49 shows domains that have been visited by the current user.

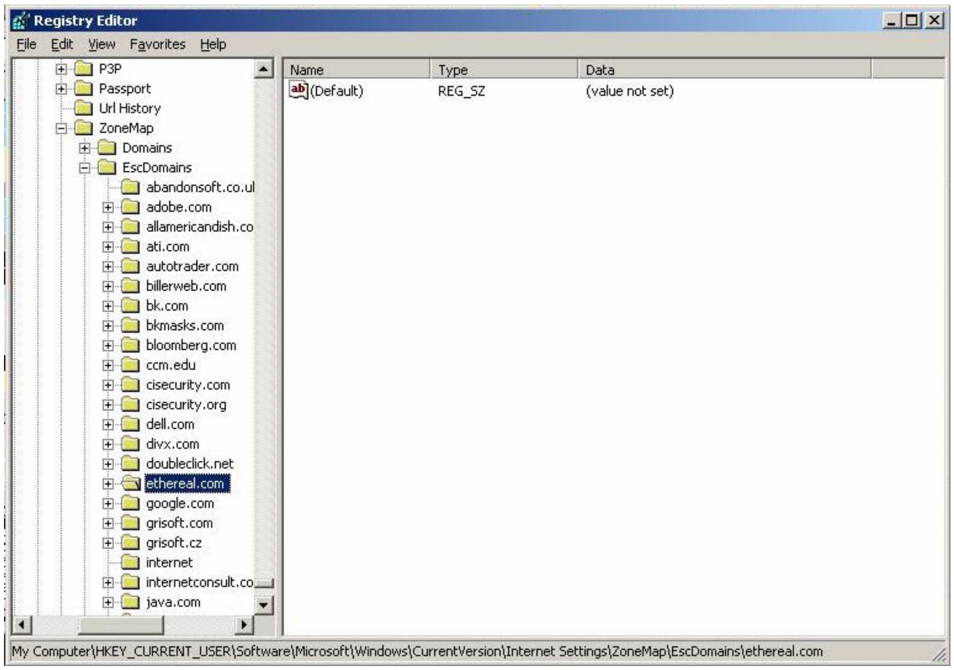


Figure 49 Domains

Figure 50 shows the executable that gets installed as a result of a user downloading and installing Ethereal from the Internet, including the physical path of the file.

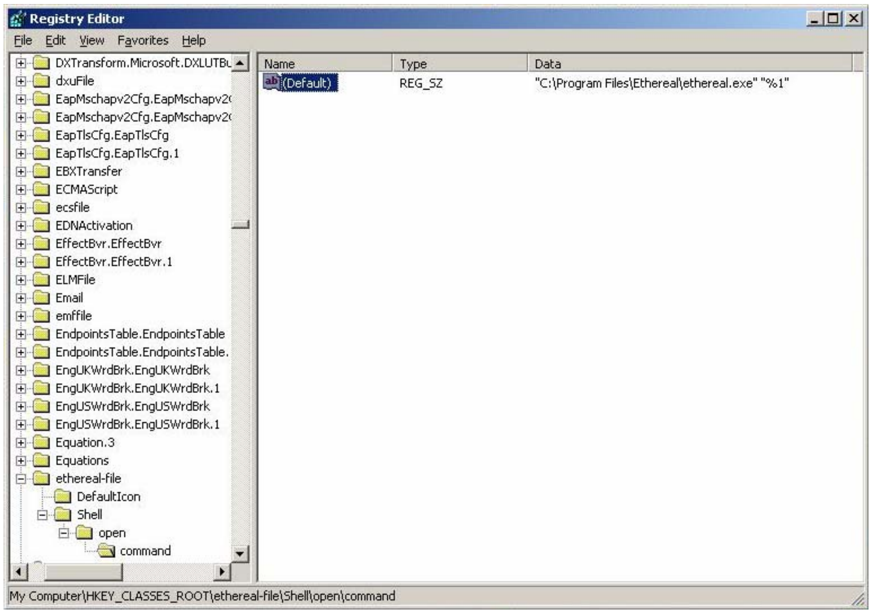


Figure 50 Executable

Figure 51 shows where the registry records the execution of both the Ethereal installer and the Ethereal program itself.

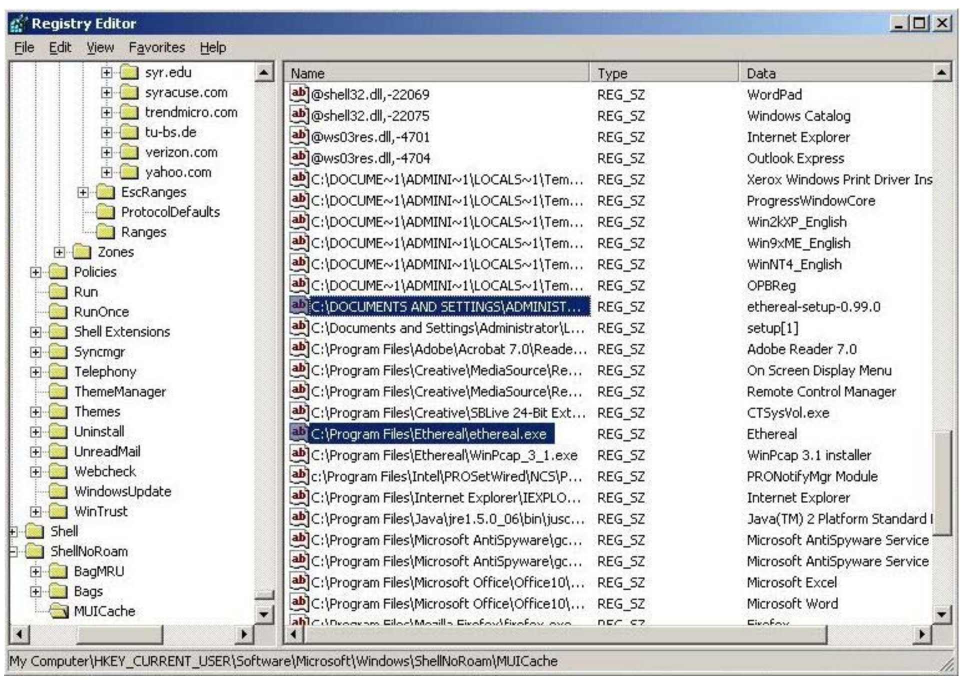


Figure 51 Execution of Installer and Program

Figure 52 shows the Ethereal installation among the other programs and executables present on the system.

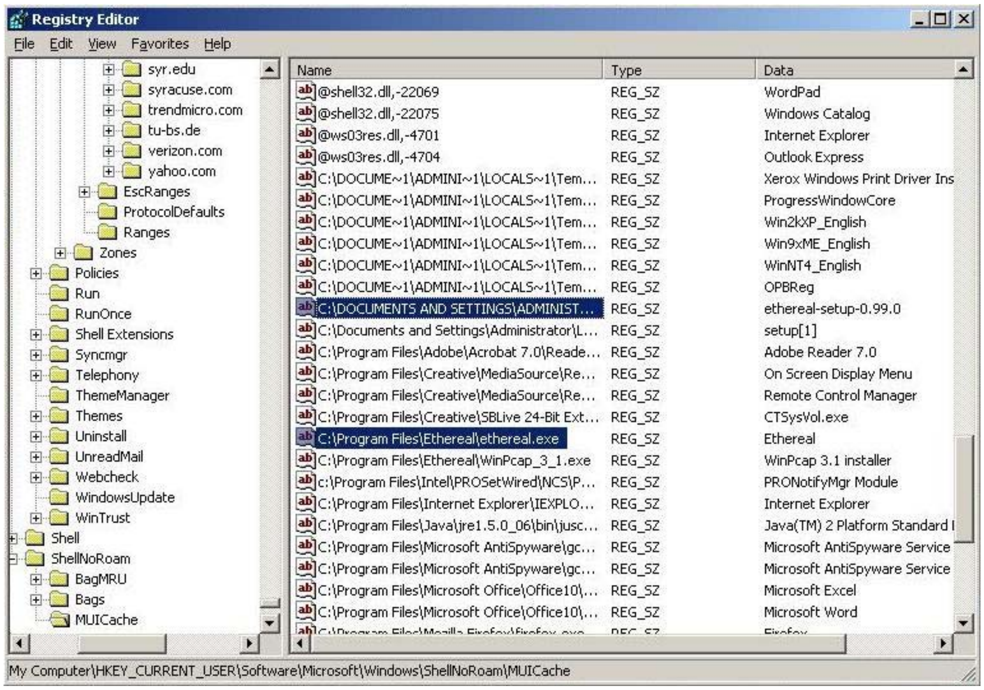


Figure 52 Installation Record

Finally, uninstall information for the program is also stored in the registry, as depicted in Figure 53. Generally if a program is malicious in nature (Ethereal is not) then the information given in this section may be false or misleading.

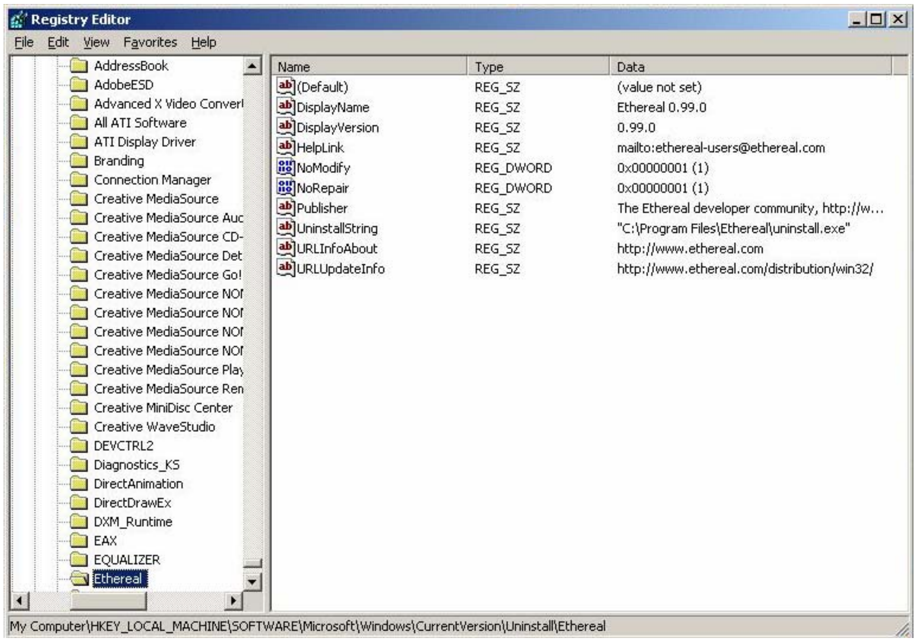


Figure 53 Uninstall Information

It should be noted that all of the information captured in the registry regarding the downloading and installation of Ethereal is recorded in real-time on the system and was captured immediately after the events had occurred.

D.7.2 Supplemental Scenario #2 - Regular User Leaking Info

The user plugs in USB Drive and copies over documents from the system, such as “Registry.doc” in this example.

This scenario is hard to detect using the Windows Registry alone. The key is to correlate the mounting of a removable drive with the access of sensitive documents by the current user. In the first figure, the Windows Registry detects a removable drive mounted on H:\.

Figure 54 shows a removable storage drive mounted on the system by the current user.

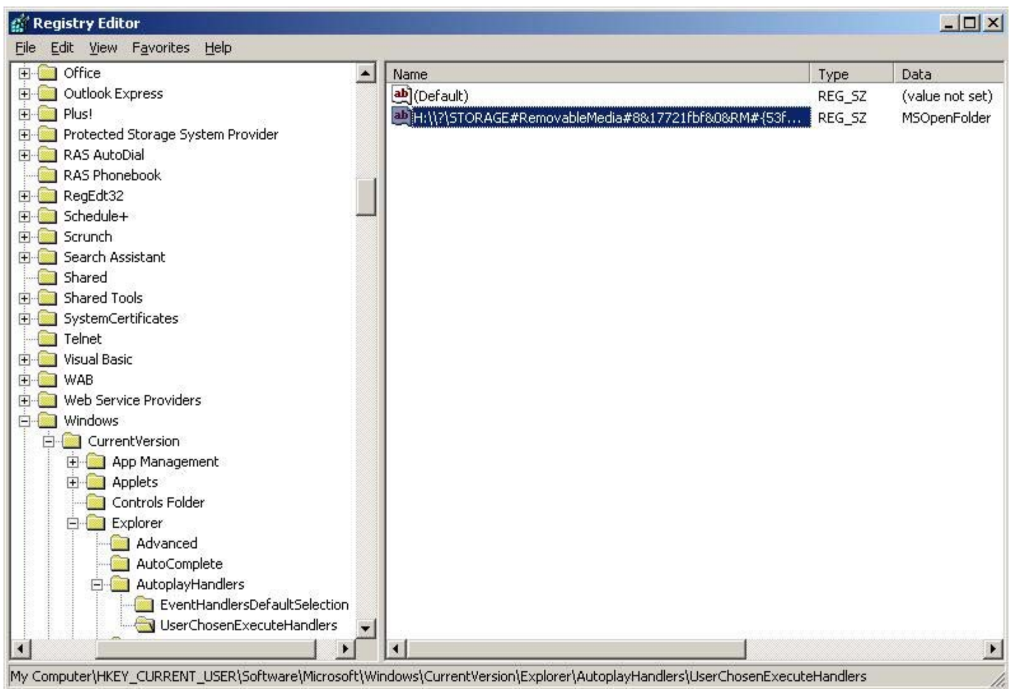


Figure 54 Record of Removable Storage

Figure 55, the Windows Registry identifies any documents that have been accessed by the current user.

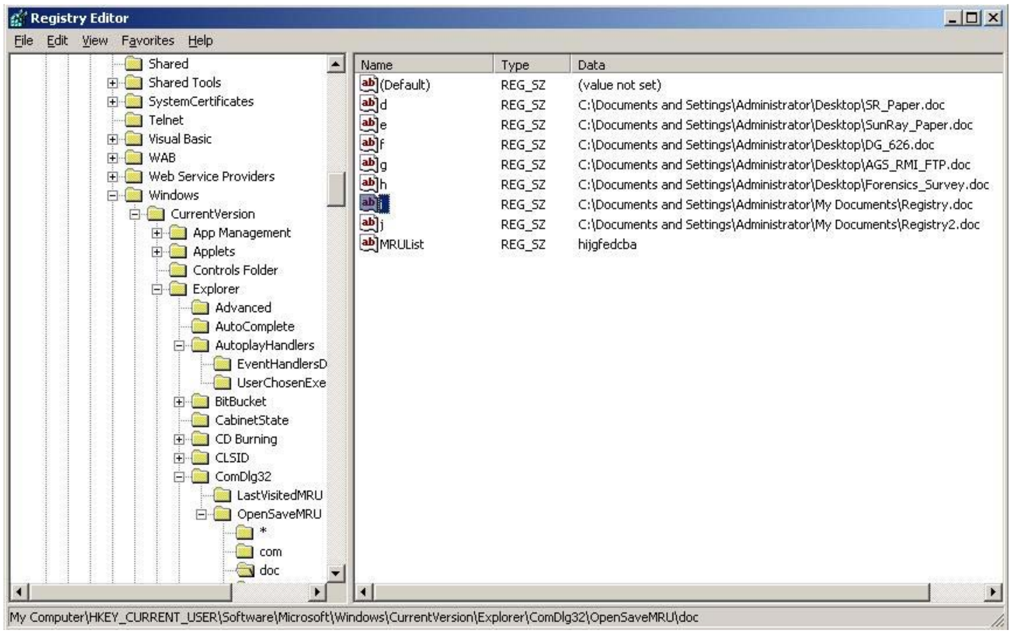


Figure 55 Accessed Documents

Given this information, it is technically feasible that the current user saved any of the intellectual property above to a removable drive. The documents in question could be spreadsheets, databases, diagrams, and other possibly sensitive or confidential information.